



HAL
open science

Quelque contribution à l'optimisation de constellations de satellites

Grandchamp Enguerran

► **To cite this version:**

Grandchamp Enguerran. Quelques contributions à l'optimisation de constellations de satellites. Informatique [cs]. Institut National Polytechnique de Toulouse - INPT, 2001. Français. NNT: . tel-00634769

HAL Id: tel-00634769

<https://theses.hal.science/tel-00634769>

Submitted on 23 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numéro d'ordre : 1844

Thèse

présentée

pour obtenir le titre de

**DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE
TOULOUSE**

Spécialité : **INFORMATIQUE**

par

Enguerran GRANDCHAMP

**QUELQUES CONTRIBUTIONS POUR L'OPTIMISATION DE
CONSTELLATIONS DE SATELLITES**

thèse soutenue à l'ENSEEIHHT le 17 décembre 2001 devant le Jury composé de :

	Président
Jin-Kao Hao	Rapporteurs
Jean-Marc Alliot	
Joseph Noailles	Directeur de thèse
Pascal Brousse	Examineurs
Vincent Charvillat	
Pierre Dago	

Résumé

Quelques contributions pour l'optimisation de constellations de satellites

A mi-chemin entre l'optimisation et l'astronautique, ce travail de recherche s'attaque au problème du design de constellations de satellites. Déterminer le nombre de satellites nécessaires pour répondre à un besoin et les positionner de manière correcte, tels sont les enjeux techniques de cette thèse. Minimiser les coûts et les délais, tels sont les enjeux économiques auxquels est confronté le domaine spatial au quotidien.

Les difficultés majeures de ce problème sont : d'une part la taille et les caractéristiques de l'espace de recherche; D'autre part la non-régularité des critères utilisés et la nature mathématique et physique hétérogène de l'ensemble des paramètres (paramètres entiers et paramètres réels) interdisant l'utilisation d'algorithmes classiques; Enfin l'évaluation des solutions qui met en oeuvre une simulation coûteuse en temps et en calculs sans retourner d'information pertinente sur les bonnes, ou mauvaises, propriétés de la constellation évaluée interdit une exploration massive de l'espace de recherche.

De ces constatations et de diverses études préliminaires est née une nouvelle approche. Basé sur une meilleure exploitation de la simulation et sur une simplification des critères utilisés, l'algorithme composé de plusieurs niveaux met en oeuvre plusieurs techniques d'optimisation. En intégrant une base de connaissance sur les orbites et une méthode numérique de recherche d'optimum, le tout orchestré par un algorithme métaheuristique, cette nouvelle approche tente de résoudre les difficultés intrinsèques au domaine d'application en décomposant le problème.

MOTS - CLEFS

- Astronautique
- Design de constellations
- Optimisation
- Métaheuristiques
- Base de connaissance
- Simulation

Abstract

Some contribution to satellite constellations optimization

Half-way between optimization and astronautic, this research study deals with satellites constellations design problems. In order to find the required number of satellites and to correctly set their position, such are the technical challenges of this thesis. To minimise the cost and reduce the time such are the economical challenges the space industries are confronted in a daily manner.

The major difficulties of these problems are : the size and the characteristics of the search space; The irregularity of the criterions; The mathematical and physical heterogeneity of parameters forbids the use of classical algorithms; The evaluation of a solution, which uses a time consuming evaluation without returning pertinent information about the good or bad properties of the constellations, forbid a massive exploration of the search space.

From this constatation and several preliminary studies a new approach is born. Based on a better use of the simulation and on a simplification of the criterions, the algorithm is composed of several levels and uses different optimization techniques: it integrates a knowledge database on the orbits and a numerical optimization process both orchestrated by a metaheuristic algorithm. This new approach tries to bypass the main drawbacks of the field with a decomposition of the problem.

KEYWORDS

- Astronautic
- Constellation design
- Optimization
- Metaheuristics
- Knowledge data base
- Simulation

Remerciements

Je remercie les membres du jury

- Monsieur Joseph Noailles, professeur à l'INP-ENSEEIH, président de ce jury et également mon directeur de thèse pour avoir encadré mon travail.
- Messieurs Jin-Kao Hao, Professeur à l'université d'Angers, et Jean-Marc Alliot, Professeur à l'ENAC, pour avoir accepté de juger ce travail en tant que rapporteurs spécialistes des techniques d'optimisation métaheuristiques.
- Messieurs Pascal Brousse, chef de service chez Alcatel Space Industries, et Pierre Dago, ingénieur chez Alcatel Space Industries, pour avoir accepté d'examiner ce travail.
- Monsieur Vincent Charvillat, maître de conférence à l'INP-ENSEEIH, qui a encadré mon travail avec beaucoup de présence et d'efficacité.

Je remercie les membres de l'ENSEEIH avec qui j'ai travaillé pendant quatre ans. La liste étant longue je me limiterai aux personnes directement impliquées dans mon travail doctoral (recherche, rédaction du manuscrit). Mais je n'oublie pas tous les collègues avec qui j'ai travaillé en enseignement et qui m'ont témoigné beaucoup de sympathie. Je remercie donc

- Max Buvry pour sa grande maîtrise de LateX très utile pendant la rédaction.
- Vincent Charvillat, que je cite à nouveau et que je pourrais citer à chacune des pages de ce manuscrit tant son aide m'a été précieuse et bénéfique. Je souligne par là ses qualités de synthèse et de lucidité fort utiles durant les phases critiques et difficiles de ce travail mais également son amitié et son soutien.
- Bernard Thiesse, pour son aide sur les techniques d'optimisation.

Je remercie l'ensemble du service *PML* du département *DOS/AS* d'*Alcatel Space Industrie* représenté par Erick Lansard, chef de département, Jean-Luc Palmade et Pascal Brousse chef de services successifs. Je remercie également Béatrice Deguine et Pierre Dago qui ont successivement encadré mon travail.

Je remercie Emmanuelle, Mathilde et Benjamin qui ont subi mon manque de disponibilité durant la rédaction de ce précieux manuscrit.

Notations

Les paramètres spatiaux	Re	Rayon équatorial de la terre (6378.400km)
	μ	Constante de gravitation ($398600.47.10^9 m^3.s^{-2}$)
	J_2	Dérive liée à la rotation de la terre ($1.08262583.10^{-3}$)
	$w_T(t)$	Angle de rotation de la terre entre l'instant t_0 de référence et l'instant t
Les paramètres orbitaux	a	Semi-grand axe de l'ellipse définissant l'orbite
	e	Excentricité de l'ellipse définissant l'orbite
	i	Inclinaison du plan orbital par rapport à l'équateur
	ω	Argument du périégée
	Ω	Longitude du noeud ascendant
	M	Anomalie moyenne du satellite sur son orbite
	h	altitude du satellite au périégée ($h = a(1 - e) - Re$)
Les orbites	<i>LEO</i>	Low Earth Orbit (Orbite basse)
	<i>MEO</i>	Medium Earth Orbit (Orbite moyenne)
	<i>HEO</i>	Highly Eccentric Orbit (Orbite très elliptique)
	<i>GEO</i>	Geostationary Orbit (Orbite géostationnaire)
	<i>IGSO</i>	Inclined Geostationary Orbit (Orbite géostationnaire inclinée)
Les constellations	<i>GPS</i>	Global Positioning System (opérationnelle en 1994)
	<i>GNSS</i>	Global Navigation Satellite System
	<i>GLONASS</i>	GLObal NAvigation Satellite System
	<i>INES</i>	Inovative European Navigation System

Les techniques d'optimisation	<i>GA</i>	Genetical Algorithm (Algorithmes génétiques)
	<i>SD</i>	Steepest Descent (Recherche de plus grande pente)
	<i>TS</i>	Tabu Search (recherche tabou)
	<i>OLS</i>	Ordinary Least Squares (Régression aux moindres carrés ordinaires)
	<i>ODR</i>	Orthogonal Distance Regression (Régression en distance orthogonale)
L'algorithme	<i>ODB</i>	Orbit Data Base (Base de données sur les orbites)
	<i>NM</i>	Niveau Métaheuristique
	<i>NH</i>	Niveau Heuristique
	<i>NN</i>	Niveau Numérique)
Les critères	<i>CV</i>	Constellation Value
	<i>DOP</i>	Dilution Of Precision
La modélisation		
	<i>Sat</i>	Un satellite
	<i>Cs</i>	Une constellation
	<i>Cl</i>	Une classe d'orbite
	<i>Cf</i>	Une configuration
	<i>n</i>	Nombre de satellites de la constellation
	<i>n_{min}</i>	Nombre minimal de satellites de la constellation
	<i>n_{max}</i>	Nombre maximal de satellites de la constellation
	<i>m</i>	Nombre de classes de la base de données
	Δ	Liste des configurations visitées
	Π_{Cf}	Ensemble des constellations visitées au sein de la configuration <i>Cf</i>
	LT_{Cf}^1	Liste des dernières configurations visitées
	<i>k</i>	Taille de la liste LT_{Cf}^1
	LT_{Cf}^2	Liste des configurations de mauvaise qualité
	LT_{St}	Liste des derniers satellites ajoutés
	<i>k'</i>	Taille de la liste LT_{St}
	<i>Freq</i>	Fréquence d'utilisation d'une classe
	<i>N_{os}</i>	Nombre d'itérations des oscillations stratégiques
	S_p^1	Seuil sur la progression de la valeur d'une configuration
	S_p^2	Seuil sur la progression de la composition d'une configuration

les évaluations	$V_{s1}, V_{s2}, V_{s3}, V_{s4}$	Valeurs associées à un satellite.
	V_{st}	Valeur d'un satellite (attribut qualité) calculé en fonction des quatre valeurs précédentes.
	f	fonction d'évaluation d'une constellation
	V_{cf}^1	Valeur à posteriori d'une configuration
	V_{cf}^2	Valeur à priori d'une configuration
	V_{cl}	Valeur d'une classe

Introduction

Contexte de l'étude

Cette étude a été réalisée en collaboration avec *Alcatel Space Industries* qui a financé ce travail doctoral. L'essentiel des travaux ont été menés au sein du service PML, successivement dirigé par Jean-Luc Palmade et Pascal Brousse, du département DOS/AS dirigé par Erick Lansard. Cette étroite collaboration a permis d'approcher le problème dans un contexte économique et technique industriel, avec les contraintes que cela impose, et a donné lieu à la rédaction d'un certain nombre de rapports internes dont certains sont *confidentiels société*. La liste par ordre chronologique est :

- Optimisation De Constellations De Satellites : Première Approche (Octobre 1998),
- Bibliographie Sur Les Méthodes d'Optimisation (Décembre 1998),
- Étude des doublets de satellites (Décembre 1999),
- Étude de FlexToolGA Partie I : évaluation (Juin 1999),
- Étude de FlexToolGA Partie II : étude du design de constellation (Juin 1999),
- Étude de FlexToolGA Partie III (Juin 1999),
- Étude de FlexToolGA Partie IV : problème de déploiement (Juin 1999),
- Modélisation mathématique des critères (Septembre 1999),
- Première Modélisation (Septembre 1999),
- Étude de la méthode de Hanson (Janvier 2000),
- Modélisation Et Algorithme De Résolution (Mars 2000),
- Classification Des Orbites (Mars 2000),
- Optimisation De Constellations Par Approche Géométrique (Juin 2000).

Plan du document

Le présent document est décomposé en plusieurs parties regroupant des courts chapitres portant sur le même thème.

La première partie présente le problème que nous souhaitons résoudre dans son contexte général. Cette présentation permet de comprendre les enjeux économiques et techniques du design de constellations et de mieux comprendre la place que tient cette phase dans la définition du système global (chapitre 1). Nous veillerons à bien délimiter le système que nous étudierons en détaillant ses paramètres et ses contraintes (chapitre 2). Des premières indications sur les directions à éviter pour résoudre le problème seront alors données.

Au cours de la partie suivante, nous détaillerons les différents critères utilisés pour optimiser les constellations de satellites en présentant dans un premier temps les grands principes de l'évaluation et notamment la simulation (chapitre 3), avant de nous intéresser au calcul des critères. Nous détaillerons ces calculs en fonction des trois grands domaines d'application des constellations : les télécommunications et l'observation de la terre (chapitre 4) et la navigation (chapitre 5) en incluant la prise en compte des pannes. Cette seconde partie complètera la précédente en nous fournissant une mine d'informations permettant de choisir ou de ne pas choisir certaines techniques d'optimisation.

Après avoir sommairement évoqué les méthodes d'optimisation au cours des deux précédentes parties, et après avoir fixé le cadre général dans lequel nous nous situons pour l'optimisation (chapitre 6) nous présenterons plus en détail trois algorithmes : les algorithmes génétiques (chapitre 7), la recherche tabou (chapitre 8) et la recherche de plus forte pente (chapitre 9). Les deux premières méthodes sont des algorithmes métaheuristiques, la dernière est une méthode classique d'optimisation.

Dans la quatrième partie, le coeur de ce travail de recherche sera ensuite présenté au travers de la modélisation du problème. Nous y détaillerons l'état de l'art (chapitre 10) en matière d'optimisation de constellations avant de faire un bilan sur l'ensemble des difficultés que nous avons identifiées et qui nous ont guidés vers une nouvelle approche du problème (chapitre 11). Cette approche est basée sur un algorithme en plusieurs niveaux et sur une décomposition du critère d'évaluation en vue de le simplifier. Nous détaillerons chacun des niveaux (chapitres 12 et 13) en les présentant selon le formalisme des algorithmes présentés de manière théorique dans la partie III.

Nous illustrerons ensuite l'intérêt de l'algorithme par des applications dans le domaine des télécommunications (chapitre 15) et de la navigation (chapitre 16). Les tests réalisés permettent de rendre compte des performances des différents niveaux de manière indépendante avant de les intégrer tous dans la résolution d'un problème.

Enfin nous terminerons ce document en présentant les perspectives et conclusions de cette étude. Le lecteur trouvera également en annexe (partie VI) quelques détails sur les méthodes proposées en perspective (annexe A) et sur la structure logiciel développée (outils, interfaces, ...) (annexe B).

Table des matières

Résumé	3
Abstract	5
Remerciements	7
Notations	9
Introduction	13
Partie I : Le contexte spatial	26
1 Présentation	29
1.1 Introduction	29
1.2 Les différents domaines d'application	30
1.2.1 Les télécommunications	31
1.2.2 L'observation de la terre	32
1.2.3 La navigation	33
1.3 Les différentes phases de la mission	34
1.3.1 Analyse de la mission	34
1.3.2 Le design de la constellation	34
1.3.3 La mise à poste des satellites	35
1.3.4 Contrôle des satellites	36
1.3.5 Maintien à poste des satellites	36
1.4 Bilan du système	37
1.5 Conclusion	38

2	Le système	41
2.1	Introduction	41
2.2	Le système étudié	41
2.3	Les paramètres définissant un satellite	42
2.4	Nature physique des paramètres	43
2.5	Influence des paramètres sur l'orbite	44
2.5.1	Influence de a	44
2.5.2	Influence de e	44
2.5.3	Influence de i	44
2.5.4	Influence de ω	45
2.5.5	Influence de Ω	45
2.5.6	Influence de M	45
2.5.7	Influence sur les performances de la constellation	45
2.6	La nature mathématique des paramètres	46
	Conclusion	47
	Partie II: Les critères	49
3	Les grands principes de l'évaluation	51
3.1	La propagation des satellites	51
3.1.1	Introduction	51
3.1.2	Mécanisme	51
3.2	Les systèmes de coordonnées	52
3.2.1	Le repère géocentrique (R_{geo})	53
3.2.2	Le repère galiléen (R_{gal})	54
3.3	Équations de mouvement du satellite	54
3.3.1	Équation dans le plan orbital	54
3.3.2	Équation dans le repère géocentrique	56
3.4	La simulation	57
3.4.1	Échantillonnage du temps et de l'espace	57
3.4.2	Notion de trace au sol	58
3.4.3	La durée de simulation	58
3.4.4	Les post-traitements	60
3.5	Visibilité des satellites	61
3.5.1	Calcul du cercle de visibilité	61

3.5.2	Bande de visibilité	61
3.5.3	Cas d'instruments optiques	61
3.5.4	Cas des instruments radars	62
3.6	Bilan	63
4	Critères dédiés aux télécommunications et à l'observation	65
4.1	Introduction	65
4.2	Critère dédié aux télécommunications	66
4.2.1	Principe	66
4.2.2	Les post-traitements	67
4.3	Critère dédié à l'observation	67
4.3.1	Présentation	67
4.3.2	Expression de la fonction critère	68
5	Calcul du critère dans le cas de la navigation	69
5.1	Introduction	69
5.2	Principe de triangulation	70
5.3	Détail des calculs	72
5.4	Évaluation du critère d'optimisation	74
5.4.1	Introduction	74
5.4.2	Les différents critères utilisables	74
5.4.3	Expression du critère en fonction des données de l'algorithme	75
5.4.4	Configuration optimale	76
5.5	Intégration des pannes	77
	Conclusion	79
	Partie III : Les méthodes d'optimisation utiles dans la suite de l'étude	81
6	Introduction	83
7	Les algorithmes génétiques	87
7.1	Introduction	87
7.2	Notations et principe	88
7.3	Les règles d'évolution	89
7.3.1	La sélection	89
7.3.2	Le croisement	89

7.3.3	La mutation	90
7.4	La mise en oeuvre des algorithmes génétiques	91
7.4.1	Le codage des individus et les paramètres de l'algorithme	91
7.4.2	La convergence	91
7.4.3	La condition d'arrêt	91
7.4.4	L'algorithme général	92
7.5	Aspects avancés des algorithmes génétiques	92
7.5.1	Fonction de pénalisation	92
7.5.2	La théorie des schémas	93
7.5.3	Le scaling	94
7.5.4	Notion d'élitisme-tournoi	95
7.5.5	Le sharing	96
7.6	Autres méthodes	96
8	La recherche Tabou	99
8.1	Introduction	99
8.2	Définitions	100
8.2.1	Transition avec attributs	100
8.2.2	Voisinage	101
8.2.3	Les attributs tabous	101
8.3	La mémoire	101
8.3.1	Principe	101
8.3.2	La mémoire à court terme	102
8.3.3	La mémoire à long terme	102
8.4	Les phases de la recherche	103
8.4.1	Mise en oeuvre de l'intensification	103
8.4.2	Mise en oeuvre de la diversification	103
8.5	Aspects avancés	104
8.5.1	La liste de candidats	104
8.5.2	Les critères d'aspiration	104
8.5.3	La recherche tabou probabiliste	104
8.5.4	Les oscillations stratégiques	104
8.6	Algorithme de recherche tabou	105
9	La descente de plus grande pente	107
9.1	Introduction	107
9.2	Méthode d'optimisation par direction de recherche	107

9.2.1	Détermination de la direction de recherche	108
9.2.2	Détermination du pas de recherche	108
9.2.3	Convergence	108
9.3	La méthode de recherche de plus forte pente	108
9.3.1	Recherche de la direction	108
9.3.2	Détermination du pas de descente	109
9.3.3	Convergence	109
Partie IV : Modélisation et résolution		111
10	État de l'art	113
10.1	Introduction	113
10.2	Présentation des algorithmes dédiés au design	113
10.2.1	Principe	113
10.2.2	Exemple : la méthode <i>SOC</i>	114
10.2.3	Exemple : la méthode de Walker	114
10.2.4	Bilan	117
10.3	La méthode de Hanson	117
10.3.1	Introduction	117
10.3.2	Principe de la méthode	117
10.3.3	Nombre de passages au dessus du point d'intérêt	119
10.3.4	Influence des paramètres i , h et Ω	121
10.3.5	Adaptation et automatisation de la méthode	121
10.3.6	Le critère	122
10.3.7	Généralisation au design de constellations	123
10.3.8	Un nouveau problème	124
10.3.9	Généralisation au cas de plusieurs zones d'intérêt	126
10.3.10	Application de notre algorithme	126
10.3.11	Bilan	127
10.4	Les algorithmes génétiques (AG)	127
10.4.1	Pourquoi les AG	127
10.4.2	Les limitations	128
10.4.3	Un opérateur de croisement inadapté	128
10.4.4	Bilan	129
10.5	Autres algorithmes évolutifs	129
10.6	Conclusion	130

11 Modélisation : vers une nouvelle approche	133
11.1 Introduction	133
11.2 Justification de notre approche	134
11.2.1 Un problème non standard	134
11.2.2 Un espace de recherche complexe	134
11.2.3 Un critère coûteux et irrégulier	135
11.2.4 Une recherche pas trop aveugle	135
11.3 Modélisation et résolution proposée	136
11.3.1 Principe de décomposition	136
11.3.2 Principe de résolution	137
11.3.3 Bilan modélisation-résolution	137
11.4 Décomposition de l'espace de recherche	138
11.4.1 Introduction	138
11.4.2 Notion de classe d'orbites	138
11.4.3 Notion de configuration	139
11.5 Classes d'orbites définies par un intervalle	139
11.5.1 Principe	139
11.5.2 Notion d'intervalle flou	140
11.5.3 Centre et bordure de la classe	141
11.5.4 Égalité entre deux classes	141
11.5.5 Distance entre deux classes	141
11.6 Exemple de classes d'orbites définies par intervalle	141
11.6.1 Étude des orbites circulaires	142
11.6.2 Étude des orbites à haute-altitude	145
11.6.3 Étude des orbites basses et des orbites moyennes excentriques	145
11.7 Paramètres des classes génériques	146
11.7.1 Conclusion	147
11.8 Classe définie par une relation entre des paramètres	147
11.8.1 Principe	147
11.9 Odb : fonctionnalités	148
11.10 Les configurations	148
11.10.1 Introduction	148
11.10.2 Distance entre deux configurations	149
11.10.3 Relation d'équivalence	149
11.10.4 Relation d'ordre	149
11.10.5 Remarques	149

11.11	Quelques remarques	149
11.12	Conclusion	150
12	Résolution : le niveau numérique	151
12.1	Introduction	151
12.2	Principe	152
12.3	Mise en oeuvre de la descente de plus grande pente	153
12.3.1	Définition du voisinage	153
12.3.2	Amplitude de la perturbation	154
12.3.3	Justification	154
12.4	Rencontre d'optimum locaux	155
12.4.1	Définition de l'optimum local	155
12.4.2	Principe de redémarrage	156
12.5	Durée de la recherche	156
12.6	Bilan	156
13	Résolution : le niveau métaheuristique	159
13.1	Introduction	159
13.2	L'espace de recherche	160
13.2.1	Définition	160
13.2.2	Évaluation de la complexité de l'espace	161
13.3	Les transitions	162
13.4	Le voisinage	163
13.4.1	Définition	163
13.4.2	Taille du voisinage	164
13.5	De multiples critères	164
13.5.1	Évaluation d'une constellation	165
13.5.2	Évaluation d'un satellite	165
13.5.3	Évaluation à posteriori d'une configuration	166
13.5.4	Évaluation d'une classe	166
13.5.5	Évaluation à priori d'une configuration	166
13.5.6	Bilan	167
13.6	La mémoire	167
13.6.1	La récence	167
13.6.2	La qualité	168
13.6.3	La fréquence	169
13.6.4	Autres structures	169

13.6.5	Bilan	169
13.7	Mise à l'état tabou des configurations	169
13.7.1	Utilisation de LT_{cf}^1	170
13.7.2	Utilisation de l'attribut <i>Freq</i> des classes	170
13.7.3	Intégration de l'aspect qualité de la mémoire	170
13.7.4	Notion de borne tabou	171
13.8	Choix du successeur	171
13.8.1	Principe	171
13.8.2	La liste de candidats	172
13.8.3	Les critères d'aspiration	172
13.9	Application des transitions	172
13.9.1	Application d'une transition de type T_0	173
13.9.2	Application d'une transition de type T_1	173
13.9.3	Application d'une transition de type T_2	173
13.9.4	Retour vers une configuration déjà visitée	174
13.10	Les Oscillations stratégiques	175
13.10.1	Principe	175
13.10.2	Évaluation de la progression d'une solution	176
13.10.3	Que faire si le problème a été satisfait	176
13.11	Conclusion	177

Intégration et conclusion **179**

Partie V : Application **183**

14 Évaluation de l'algorithme de bas niveau **185**

14.1	Évaluation de l'algorithme de descente	185
14.1.1	Conditions des tests	185
14.1.2	Tableaux	186
14.1.3	Interprétation	187
14.2	Évaluation du redémarrage	188
14.2.1	Conditions des tests	188
14.2.2	Tableaux	189
14.2.3	Interprétation	189
14.3	Tests approfondis	189
14.3.1	Conditions des tests	189

14.3.2	Tableaux	190
14.3.3	Interprétation	190
15	Application en télécommunications	193
15.1	Introduction	193
15.2	Remarques préliminaires sur les techniques d'évaluation	193
15.2.1	Test du niveau numérique	193
15.2.2	Test du niveau métaheuristique	194
15.2.3	Test de l'algorithme global	194
15.3	Recherche du nombre minimal de géostationnaires	195
15.3.1	Test du niveau numérique	196
15.3.2	Test du niveau métaheuristique	198
15.3.3	Intégration des différents niveaux	203
15.4	Premières conclusions	204
16	Applications en navigation	207
16.1	Introduction	207
16.2	Ajout de 6 satellites à la constellation de Walker 18/6/1	207
16.2.1	Test du niveau numérique	208
16.2.2	Test du niveau métaheuristique	211
16.2.3	Intégration des différents niveaux	213
16.3	Ajout de 6 satellites à la constellation <i>GPS</i>	215
16.3.1	Résultats	215
16.4	Extension au cas de n satellites	216
16.4.1	Principe	216
16.4.2	Résultats	217
16.5	Extension du nombre de classes	218
16.5.1	Principe	218
16.5.2	Test du niveau numérique	219
16.5.3	Test global	219
16.6	Conclusion	220
	Perspectives et conclusions	221
	Conclusions	221
	Limitations	222
	Perspectives	223

Partie VI: Annexe	227
A Estimation de paramètres	229
A.1 Introduction	229
A.2 Problème de régression aux moindres carrés ordinaires (<i>OLS</i>)	229
A.3 Régression non linéaire en distance orthogonale (<i>ODR</i>)	230
A.4 Points aberrants et points levier	230
A.5 Estimation robuste : les M-Estimeurs	231
A.6 Application à l'estimation d'orbites	231
B les outils	235
B.1 Présentation	235
B.2 Illustration	235
B.3 Méthode de Hanson	238
Références bibliographiques	239

Partie I

Le contexte spatial



La présentation du problème dans son contexte économique de baisse des coûts, et dans son contexte technique lié au gain de performance, est nécessaire pour comprendre les enjeux de l'optimisation et la place de la phase de design de la constellation dans la mise en place d'un système spatial. Cette mise en place va de la définition du domaine d'application à la "déorbitation" des satellites en fin de phase opérationnelle.

Par ailleurs, une étude détaillée des paramètres du système permet de mieux appréhender la complexité du problème : différentes natures physiques et mathématiques des paramètres; nombre important de paramètres. Cette première étude nous donnera des pistes quant aux méthodes d'optimisation à utiliser, ou à ne pas utiliser, en présentant les difficultés engendrées par ces paramètres.

Nous veillerons à bien mettre en évidence le sous système que nous allons étudier ainsi que les paramètres que nous allons optimiser. Nous détaillerons chacun des aspects cités précédemment au cours de cette partie en précisant les points qui nous ont guidés dans l'élaboration de notre approche.

Chapitre 1

Présentation

1.1 Introduction

Tels Galilée et Jules Verne, la conquête de l'espace a depuis longtemps passionné les scientifiques et les écrivains. Après la gloire des premiers astronautes comme Youri Gagarine, premier homme dans l'espace en 1961, ou Neil Armstrong, premier homme à marcher sur la lune en 1969, les possibilités stratégiques et économiques de l'espace, même restreint au voisinage de la terre (0 à $36000km$), apparurent comme infinies. Des applications diverses et variées allant de l'observation de la terre pour des prévisions météorologiques ou des stratégies militaires à la transmission de données multimédia ont germé dans la tête de nombreux scientifiques et financiers.

En raison des capacités limitées des lanceurs, les premiers satellites (Spoutnik 1 en 1957, Explorer 1 en 1958) étaient de petite taille (moins de $100kg$) et lancés en orbite très basse (moins de $400km$). Mais les besoins grandissants tant militaires que civils amenèrent les ingénieurs à concevoir des satellites de plus en plus gros (jusqu'à plusieurs tonnes). C'est notamment la conquête de l'orbite géostationnaire pour les satellites météorologiques et de télécommunication qui nécessite les plus gros systèmes de bord. La distance à la terre ($36000km$) requiert des antennes puissantes et une quantité d'ergols conséquente pour mettre les satellites sur orbite.

Avant l'apparition des lanceurs pouvant accueillir plusieurs satellites simultanément, les contraintes budgétaires ne permettaient pas d'envisager des constellations ayant plus de quelques satellites (constellation Inmarsat composée de 4 satellites). Pour optimiser la couverture des satellites dédiés aux télécommunications, les satellites en orbite haute furent les seuls envisagés. On se limita même aux satellites géostationnaires dans ce domaine, ce qui conduisit à un engorgement de cette altitude (871 satellites lancés en orbite géostationnaire) et des difficultés pour placer les satellites apparurent. Dans le domaine de l'observation, les contraintes diffèrent et la continuité du service au dessus d'une même région n'est plus un critère. Les besoins de couverture ne sont pas aussi forts et les satellites en orbite basse sont utilisés dès 1974.

Avec les progrès technologiques (lanceurs multiples, miniaturisation des composants, ...) et l'évolution des besoins (besoin de couverture continue, de robustesse, de temps de réponse plus courts, ...), on commença à envisager des constellations pouvant comporter beaucoup plus de satellites et par conséquent la baisse de l'altitude des satellites devint praticable.

Les études théoriques de J.G.Walker ([100]) dès 1970 présentent une famille de constellations ayant un certain nombre de propriétés de symétrie et permettant de rapidement dimensionner une constellation à la main. Mais il faudra attendre 1986 avant que le premier satellite de la première constellation de grand gabarit soit lancé : c'est la constellation *GPS* qui fournit un service permettant le positionnement et qui est composée de 24 satellites en orbite moyenne (20183km).

Plus récemment un certain nombre de constellations ont été mises à l'étude pour des besoins de télécommunication et notamment de multimédia. C'est dans ce cadre que la constellation Skybridge a été conçue par Alcatel Space Industries en 1998. Elle comporte 64 satellites en orbite basse (1500km). La tendance à la fin des années 90 était en effet de concevoir des constellations comportant plusieurs dizaines de satellites (64 pour Skybridge, 72 pour Iridium, 288 pour télédesic) en orbite basse (altitude inférieure à 1500km) qui offrent l'avantage de nécessiter un équipement moins performant pour des satellites plus petits et plus facilement remplaçables. Mais les contraintes économiques n'ont pas permis de rendre opérationnelles de telles constellations.

Avant de décrire la manière de concevoir et de mettre sur orbite une constellation (paragraphe 1.3), nous allons nous intéresser à leur application (paragraphe 1.2). Nous allons détailler les deux grands domaines conduisant à la définition de constellations comportant beaucoup de satellites que sont la navigation et les télécommunications. Le paramètre imposant un grand nombre de satellites est certainement le besoin de continuité du service exigé par ces systèmes.

1.2 Les différents domaines d'application

Dans le contexte économique et technologique actuel les données que nous manipulons au quotidien proviennent de sources diverses et variées. Les nouveaux moyens de télécommunication ont envahi le marché et on cherche de plus en plus à les faire interagir. Si on prend l'exemple de la figure 1.1, nous sommes en présence d'un détachement de pompiers faisant face à un feu de forêt. En utilisant un satellite de télécommunication (en orbite géostationnaire par exemple), ces pompiers peuvent alerter la caserne la plus proche ainsi que les médias. Ils signalent leur position en utilisant les données fournies par la constellation *GPS* et leur récepteur embarqué. A l'aide d'une caméra, d'un récepteur GPS et d'un hélicoptère, les médias peuvent alors se rendre sur place et filmer la scène. Ces données multimédia sont alors transmises à une station mobile avant d'être renvoyées vers le centre de diffusion de l'information via une constellation fournissant un débit suffisant pour transmettre de telles données (une constellation en orbite basse par exemple). Enfin le centre de

recherche peut souhaiter prendre quelques clichés de la zone sinistrée à plus grande échelle afin de constituer une base de données, il contacte donc un satellite d'observation tel que Spot (satellite en orbite basse) qui observe toute la région. Ces données, stockées numériquement peuvent ensuite être diffusées vers le grand public ou étudiées dans des centres de recherche.

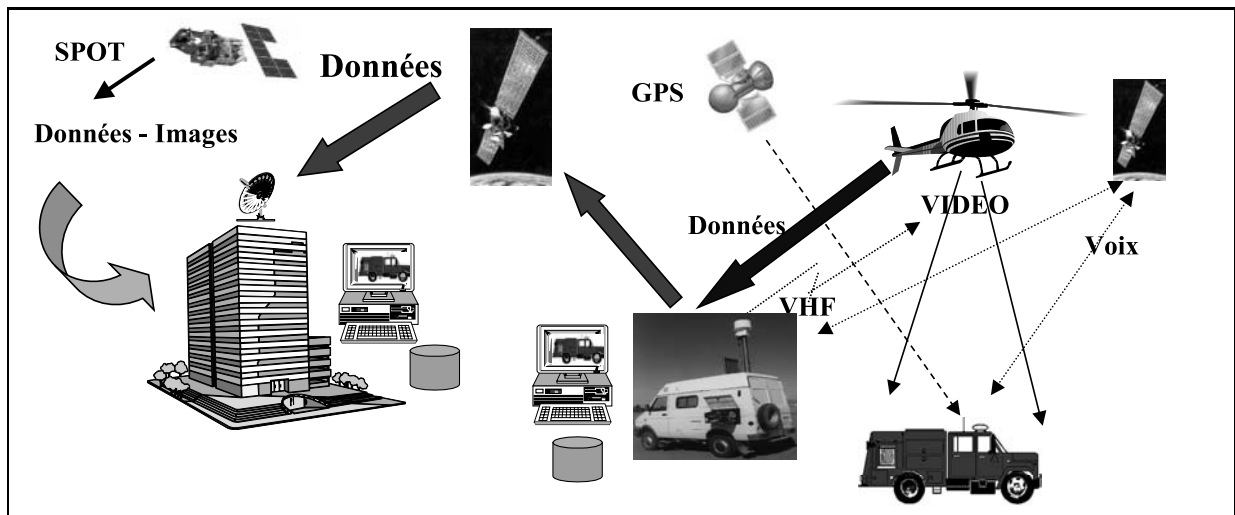


FIG. 1.1 – Exemple d'utilisation des systèmes spatiaux

Cet exemple donne une idée des domaines d'application des systèmes spatiaux. Nous allons présenter au cours de cette partie les trois principaux domaines d'application des constellations de satellites à savoir les télécommunications, l'observation et la navigation. Nous détaillerons les contraintes intrinsèques au domaine sans détailler les calculs de critères qui donneront lieu à un autre chapitre (cf. partie II page 49). Nous retiendrons que les enjeux et les contraintes de ces domaines étant différents, les constellations optimales sont de natures différentes.

1.2.1 Les télécommunications

Si on abstrait les problèmes liés au partage de bandes de fréquences ou à la bande passante fournie par la constellation, le problème sous-jacent à tout problème de télécommunication est un problème de couverture de zones géographiques.

Pour assurer le bon fonctionnement d'un système de communication il faut assurer la continuité du service dans le temps (pas de coupure d'une communication établie) et dans l'espace (pas de coupure d'une communication lors du déplacement de l'utilisateur par exemple) au dessus des zones d'intérêt (la terre entière ou des zones à forte densité de population). Pour simplifier le problème les premiers systèmes spatiaux étaient composés uniquement de satellites géostationnaires (constellation Inmarsat composée de 4 satellites géostationnaires dédiés aux télécommunications). Ce choix

permettait de réduire le nombre de satellites du fait de la haute altitude et du maintien des satellites au dessus des zones d'intérêt (L'Europe, les États-Unis, le Japon, ...).

L'inconvénient en phase opérationnelle de ce type de constellation réside dans le fait que la distance de ces satellites à la terre impose un temps de réponse relativement long (250ms pour un parcours terre-satellite) et que les systèmes embarqués (côté satellite et côté utilisateur) coûtent cher.

Mais le domaine des télécommunications est un champ d'application très important en plein essor avec notamment l'objectif de transférer des données multimédia (télévision, vidéo, sons, images) interactives.

L'émergence de ces nouveaux besoins oriente naturellement les recherches vers d'autres types de constellations. Dans ce registre, un certain nombre de constellations ont été étudiées, comme les constellations Skybridge (Alcatel) ou Teledesic (Microsoft). Mais aucune n'a encore été mise à poste pour fournir de tels services. Les seuls satellites destinés au grand public et qui permettent de diffuser de la vidéo ne le font que de manière descendante (du satellite vers les récepteurs) dans le cadre de programmes de télévision par satellite (satellites telecom 2, Eutelsat, Astra).

De manière générale, il faut dimensionner une constellation qui permette d'assurer un certain taux de couverture, dépendant des performances attendues. Nous reviendrons sur les techniques de calcul de couverture dans la partie consacrée aux critères (partie II 49).

1.2.2 L'observation de la terre

L'observation de la terre peut être réalisée pour deux raisons principales : des applications météorologiques quotidiennes; des applications de surveillance à long terme. Dans le premier cas, l'utilisation de satellites géostationnaires permet de répondre à la demande. On trouve par exemple les systèmes suivants :

- SMS 1 est le premier satellite météorologique en orbite géostationnaire, il est lancé en 1974
- 6 satellites géostationnaires de type Goes sont lancés entre 1975 et 1983,
- la constellation météorologique argos est composée de 5 satellites géostationnaires dont MétéoSat

Dans le deuxième cas, la zone d'intérêt ne se restreint pas à un pays ou un groupement de pays mais à l'ensemble de la surface terrestre. Les applications dans ce domaine sont nombreuses (études scientifiques sur l'évolution de la végétation d'un pays, télédétection de cibles, études de catastrophes naturelles, ...) et les contraintes changent radicalement. L'aspect lié à la continuité du service disparaît et est remplacé par la notion de délai de revisite maximal : on souhaite que le satellite repasse au dessus du même point avant un certain délai de temps. Cette nouvelle définition des contraintes permet de s'intéresser à des constellations comportant peu de satellites (de 1 à 5) (les satellites spot par exemple sont au nombre de 4 et fonctionnent indépendamment), mais le choix des orbites

devient plus problématique. En effet, un certain nombre de contraintes sont liées aux applications des systèmes étudiés.

- Pour une télédétection, les orbites basses sont privilégiées pour augmenter la précision des mesures. Mais on doit faire face au compromis entre la taille de la zone visible par le satellite et la résolution des prises de vue en jouant sur l'altitude (en orbite basse la couverture est moins bonne mais la résolution est meilleure alors qu'en orbite haute c'est l'inverse).
- Pour des raisons de comparaison des prises de vue, le satellite doit passer au dessus d'une même zone avec les mêmes conditions d'éclairement à chaque passage. Cela implique que l'heure de passage et l'orientation avec le soleil soient les mêmes (notion de phasage et d'héliosynchronisme).

Ces différentes contraintes entraînent des relations entre les valeurs de certains paramètres des satellites (entre l'altitude et l'inclinaison par exemple [20]). Nous détaillerons ces différentes notions dans la partie II page 49 consacrée à l'étude des critères.

1.2.3 La navigation

Le dernier grand domaine d'application des constellations de satellites est celui de la navigation. Qu'elle soit aérienne, maritime ou encore terrestre, la navigation prend toute son importance dans le contexte économique actuel. Une telle constellation est utilisée pour déterminer une position à la surface de la terre en fonction de données envoyées par les satellites.

Aujourd'hui, seule la constellation américaine GPS (Global Positioning System) ([85]), opérationnelle depuis décembre 1993, permet à n'importe quel utilisateur situé à la surface de la terre et muni d'un récepteur adapté de déterminer sa position. Du système de navigation embarqué dans les voitures à ceux embarqués dans les avions, les utilisations sont nombreuses et tendent à se développer (montres avec récepteur, systèmes pour randonneurs, ...).

Au regard des systèmes souvent critiques qui sont liés aux données transmises par les satellites, beaucoup de contraintes sont appliquées à ce domaine de manière stricte. C'est notamment le cas dans l'aviation civile et militaire. De ce fait les spécifications pour de nouveaux systèmes de navigation par satellite sont souvent mises à jour et plusieurs sont à l'étude aujourd'hui (Glonass, GNSS [47]) pour pallier la dépendance vis-à-vis du système GPS.

Parmi les contraintes que l'on rencontre, il y a la notion de précision des mesures de position. Cette précision, verticale ou horizontale, dépend de la géométrie des satellites dans le ciel de l'utilisateur. Le problème ne se résume donc plus, comme dans le cas des télécommunications, à un problème de taux de couverture garanti (qui doit être de 4 pour qu'un utilisateur détermine sa position, chapitre 5), mais il faut également une bonne géométrie relative des satellites ce qui rajoute à la complexité du problème d'optimisation.

1.3 Les différentes phases de la mission

Nous allons présenter ici les différentes phases nécessaires à la mise en place d'une constellation. Elles sont communes à toutes les missions, quel que soit leur domaine d'application, et chacune de ces phases donne lieu à une optimisation. Nous allons décrire brièvement chacune d'entre elles séparément mais il faut garder à l'esprit que ces phases sont dépendantes et que l'optimalité d'une phase n'implique pas nécessairement la faisabilité d'une autre. Ceci est notamment vrai entre la phase de design et la phase de mise à poste. Une mauvaise configuration des plans orbitaux (même optimale du point de vue des performances) peut conduire à un coût de mise à poste supérieur au maximum tolérable (en ayant par exemple un lanceur par satellite). Nous étudierons ensuite plus en détail le système auquel nous allons nous intéresser.

1.3.1 Analyse de la mission

La phase d'analyse de mission est difficile car elle doit prendre en compte de nombreux critères souvent mal définis par les clients et susceptibles de changer. Elle est en amont de toutes les autres phases et doit permettre de définir les contraintes sur les autres phases : le nombre maximal de satellites, les performances minimales, le coût maximal,

En ce qui concerne le calcul du coût de la mission, il est difficile à estimer car il fait intervenir toutes les phases de la mission. Il tient compte des paramètres techniques (nombre de satellites, technologie embarquée, ...), de paramètres d'incertitude (probabilité de panne des satellites ou des lanceurs) ou encore de contraintes politiques qui peuvent influencer le choix des lanceurs par exemple.

1.3.2 Le design de la constellation

Après la spécification du besoin (domaine d'application, performances attendues et enveloppe budgétaire), on peut s'intéresser à la phase de design de la constellation. Cette phase concerne le dimensionnement de la constellation (nombre de satellites utiles) et le positionnement des satellites. Elle est au coeur du problème auquel nous nous intéressons puisqu'elle va conditionner les performances de la constellation que l'on cherche à optimiser.

Les paramètres dont on souhaite déterminer la valeur afin d'optimiser les performances de la constellation sont d'une part le nombre de satellites qu'il faut utiliser pour résoudre le problème et d'autre part la valeur des paramètres orbitaux qui définissent chacun des satellites. On remarque que le nombre d'inconnues varie au cours du processus d'optimisation ce qui impose un certain nombre de contraintes sur les techniques de recherche à utiliser. Par exemple, l'augmentation du nombre d'inconnues (à travers l'augmentation du nombre de satellites) implique une discontinuité de la fonction d'évaluation. Tout algorithme ayant besoin d'une propriété de régularité du critère ne pourra pas être mise en oeuvre pour la résolution du problème global.

Par ailleurs, le nombre de paramètres définissant un satellite (6) génère un espace de recherche de grande taille. On a en effet $6n$ variables pour une constellation à n satellites. n est généralement de l'ordre de quelques dizaines (64 pour skybride, 66 pour iridium, 75 pour ATM-Star) mais peut atteindre plusieurs centaines (288 satellites pour la constellation télédeésic). Nous reviendrons dans le prochain chapitre sur les paramètres définissant la constellation lorsque nous détaillerons le système étudié.

1.3.3 La mise à poste des satellites

Une fois la structure de la constellation déterminée, il faut déterminer la manière optimale de mettre chacun des satellites sur son orbite opérationnelle : c'est la phase de mise à poste ([81]). Les enjeux de cette phase sont d'une part la réduction des coûts et d'autre part la réduction du temps de mise à poste.

La réduction des coûts, du fait des lanceurs uniques, se ramenait autrefois à minimiser le nombre de satellites. Depuis l'apparition des lanceurs multiples et des constellations de grande taille ce n'est plus le cas et la mise à poste nécessite l'élaboration de véritables stratégies.

Les variables de ces stratégies, véritables processus d'optimisation, sont les dates de lancement, le type de lanceur utilisé et les durées de dérive des satellites. L'ensemble de ces paramètres est lié, les dates de lancement dépendent du lanceur utilisé (délai minimum entre deux lancements), un lanceur impose une certaine capacité et un certain coût. De plus, selon la base de lancement le lanceur ne peut accéder qu'à certaines plages d'inclinaison. Enfin, le satellite n'est pas placé directement sur son orbite opérationnelle. Il passe par une orbite intermédiaire et par le jeu de dérives et de poussées, le satellite atteint son orbite opérationnelle au bout d'un certain temps. La détermination de l'orbite intermédiaire, de la durée de dérive des satellites sur celle-ci et des poussées à réaliser donne lieu à un processus d'optimisation.

Le coût de la mise à poste est induit par le choix des lanceurs utilisés (nombre de lancement au total) et par la consommation en ergols des satellites pour la mise à poste. Le temps de mise à poste est quant à lui dépendant des orbites intermédiaires et des orbites opérationnelles.

Le paramètre temporel est tout aussi important que le coût car certaines dérives des satellites peuvent prendre plusieurs mois et tout délai supplémentaire sur la durée de mise à poste entraîne une augmentation des coûts (retard dans l'ouverture d'un service). Nous ne détaillerons pas plus cette phase mais nous noterons simplement que la restriction du nombre d'orbites (inclinaison, altitude) permet de mettre en place des stratégies de lancement par paquets qui réduit à la fois les coûts et les délais de mise à poste. Ce sous-problème complexe est par ailleurs critique et une constellation, même performante, peut être rejetée si la stratégie de mise à poste est trop coûteuse. Pour plus de détails, certaines études ([51], [83]) présentent les techniques classiques d'optimisation des stratégies de mise à poste.

1.3.4 Contrôle des satellites

Une fois les satellites sur leur orbite, les problèmes ne sont pas terminés. Le bon fonctionnement de la constellation est contrôlé par des stations réparties à la surface de la terre ([14]). Bien qu'il soit réduit par rapport aux autres problèmes, le placement des stations est lui aussi soumis à l'optimisation. On souhaite ici placer au mieux (avec des contraintes territoriales évidentes liées au contexte économique et politique) les stations pour assurer un suivi des satellites et détecter au plus tôt toute anomalie.

Cette phase est maintenue durant toute la phase opérationnelle et permet une bonne application du maintien à poste des satellites.

1.3.5 Maintien à poste des satellites

Lorsque les satellites sont sur leur orbite, le contrôle de leur évolution permet de détecter toute anomalie pour la résoudre. Deux types d'interventions sont nécessaires durant la phase d'exploitation des satellites : les corrections d'orbites et le remplacement d'un satellite en panne.

Les corrections d'orbites

Durant leur phase opérationnelle, les satellites dérivent régulièrement de leur orbite initiale. Pour éviter une perte de performances liée à ces dérives, on réalise des corrections d'orbite en puisant dans les ressources en ergols des satellites. Ces mouvements, de faible amplitude, dépendent des paramètres orbitaux des satellites (vitesse de dérive, ...).

Le remplacement des satellites en cas de panne

Lorsqu'un satellite tombe en panne, il faut le remplacer pour éviter que la constellation fonctionne en mode dégradé trop longtemps et ne fournisse plus les performances attendues. Deux stratégies de remplacement concurrentes peuvent être mises en place. Le choix est fonction des contraintes sur les performances de la constellation.

- le lancement à la demande : dès qu'un satellite est en panne, on planifie un nouveau lancement. Cette stratégie a pour avantage de ne faire intervenir que les satellites dont on a besoin mais les temps de remplacement peuvent atteindre plusieurs mois en fonction des disponibilités des lanceurs et des temps de mise à poste.
- les satellites de remplacement (spares) en orbite : cette stratégie utilise des satellites mis à poste en même temps que la constellation nominale, pour remplacer plus rapidement les satellites en panne. Les satellites peuvent être placés sur la même orbite que les satellites opérationnels ou être placés sur des orbites différentes afin d'optimiser la durée de déplacement vers n'importe quel satellite de la constellation. Par ailleurs ils peuvent être actifs ou inactifs durant la phase

de fonctionnement nominale de la constellation. Le nombre de spares ainsi que leur placement est sujet à un processus d'optimisation dont les critères sont d'une part la minimisation du nombre de satellites et d'autre part la minimisation du temps de remplacement. Cette optimisation peut être réalisée en même temps que le design de la constellation ou alors dans une phase ultérieure.

Le choix de la stratégie de maintien à poste dépend du domaine d'application de la constellation ainsi que des contraintes sur les performances en fonctionnement dégradé. Pour certaines applications, notamment en navigation, les constellations doivent être suffisamment robustes pour qu'un fonctionnement en mode dégradé (après une voire plusieurs pannes) soit suffisant pour satisfaire les performances attendues. D'autres applications, comme l'observation de la terre, peuvent autoriser un fonctionnement dégradé plus long et donc autoriser un lancement à la demande (voire une fabrication à la demande).

1.4 Bilan du système

Le système global (figure 1.2) englobe beaucoup de paramètres

- Phase de design de la constellation : nombre de satellites, les six paramètres de chacun des satellites.
- Phase de mise à poste des satellites : le type de lanceur (inclinaison, capacité), l'orbite de mise à poste, la stratégie de passage de l'orbite de mise à poste à l'orbite opérationnelle (durée de la dérive, ...).
- Phase de maintien à poste des satellites : stratégie de maintien à poste (en orbite, au sol), le nombre de satellites de remplacement.

Ces éléments ne sont pas tous indépendants :

- la technologie utilisée (donc le coût) est liée à l'altitude des satellites
- la stratégie de déploiement (lanceurs, nombre de satellites par lanceur, ...) est liée à l'inclinaison, aux plans utilisés, au nombre de satellites, au nombre de satellites par plan, ...
- la stratégie de maintien ainsi que son coût sont liés à la structure de la constellation.

On trouve enfin différents types de critères :

- Le coût (design, mise et maintien à poste).
- Les performances (design).
- Le temps (mise et maintien à poste).
- Consommation en ergols (mise et maintien à poste).

1.5 Conclusion

Avant de décider la commercialisation d'une constellation, on doit estimer (simultanément ou séquentiellement) tous les points abordés précédemment. Étant donnés les coûts élevés des systèmes spatiaux, la concurrence est forte et l'optimisation des systèmes est obligatoire pour réduire les investissements.

Le problème, pris dans sa totalité, englobe trop de paramètres de natures physiques et mathématiques différentes et fait appel à des techniques d'optimisation trop différentes, pour être résolu par un unique algorithme (certains critères sont déterministes, d'autres sont stochastiques).

Par ailleurs, la définition d'une évaluation multicritère, intégrant chacune des phases, pose des problèmes de compatibilité des grandeurs que l'on associe (performances, temps, coûts, ...) et des poids respectifs que l'on affecte. De plus cela conduirait à des simulations trop importantes pour être implantées sur machine.

En outre, certains critères stratégiques et politiques sont trop subjectifs pour être modélisables.

On procède donc généralement en décomposant le problème et en optimisant chacun des sous problèmes obtenus. Cette manière de procéder ne garantit pas l'optimalité de la solution trouvée mais elle rend possible la résolution du problème. Sous certaines hypothèses on peut optimiser deux phases simultanément. Par exemple le design de la constellation et la stratégie de remplacement des satellites ou encore le design de la constellation et la stratégie de mise à poste. Dans ce cadre, cette optimisation simplifie la phase de design en essayant de déterminer la constellation optimale parmi un ensemble présélectionné (constellations de Walker par exemple).

Dans le cadre de notre étude, nous nous limiterons à la phase de design de la constellation. Ce problème, déjà très complexe (chapitre 2), intégrera sous certaines hypothèses la stratégie de remplacement (cf chapitre 5) en intégrant la notion de panne dans l'optimisation de la constellation.

Comme nous pouvons le constater, les contraintes diffèrent d'un domaine à l'autre ainsi que la forme des constellations. Parmi les trois domaines présentés ici, nous illustrerons nos algorithmes avec des problèmes issus des télécommunications (couverture continue de zones), et de la navigation (couverture continue et bonne répartition des satellites).

Le système étudié sera présenté dans le prochain chapitre.

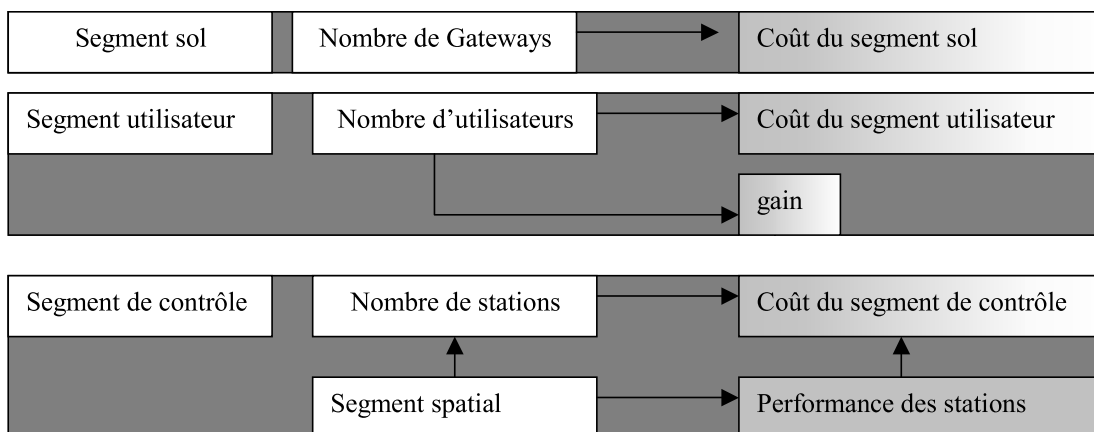
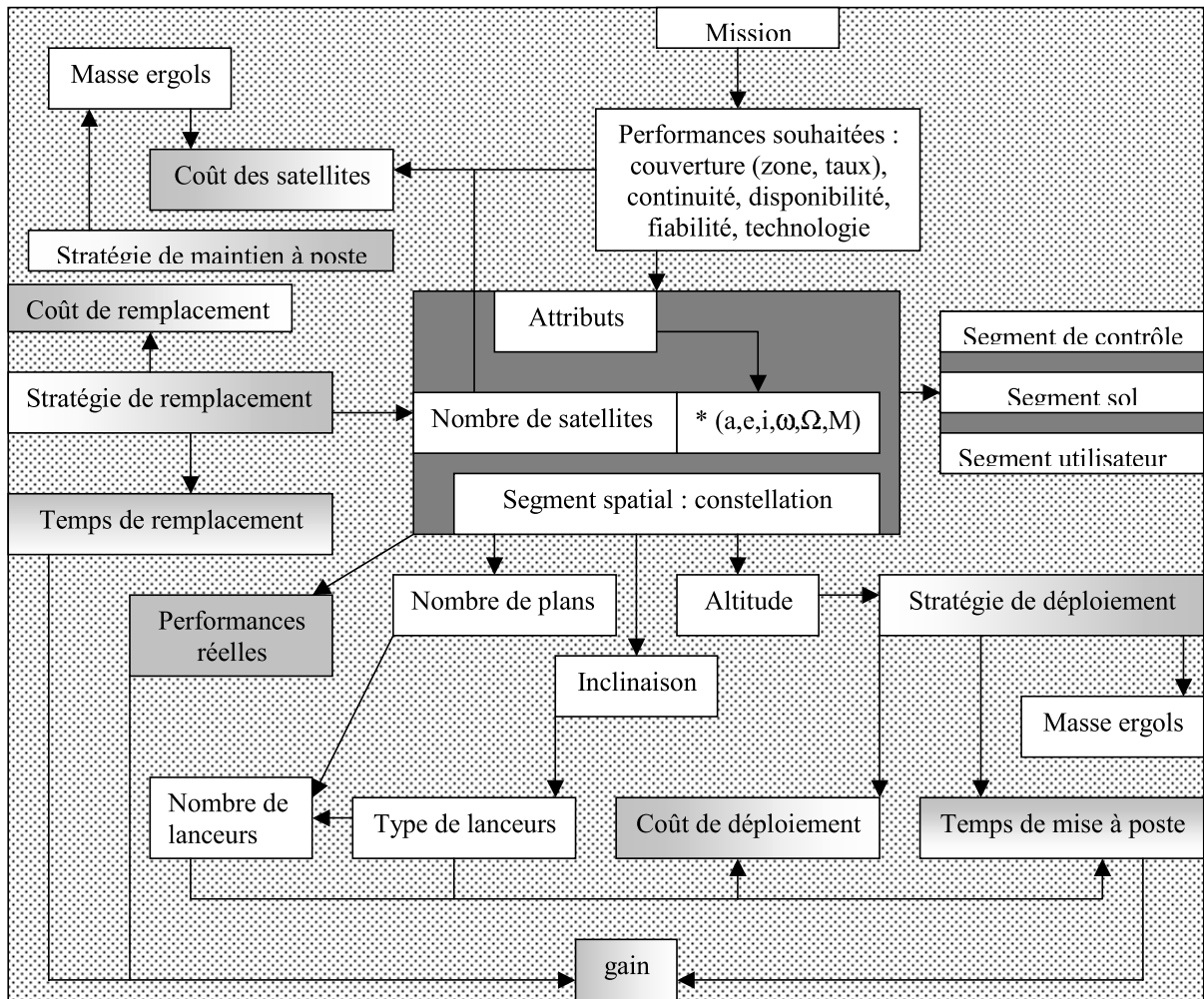


FIG. 1.2 – *Système global*

Chapitre 2

Le système

2.1 Introduction

Après la présentation de toutes les phases conduisant à la définition d'un système spatial, nous avons extrait ici le système que nous allons étudier. Cette présentation du système nous permettra de mieux appréhender un certain nombre d'obstacles à franchir pour résoudre le problème d'optimisation de constellations de satellites.

Nous allons détailler successivement au cours de ce chapitre :

- les limites du système étudié,
- les paramètres intervenant au cours de cette phase en précisant leur nature physique et mathématique
- les difficultés engendrées.

2.2 Le système étudié

Si on veut réaliser une optimisation globale, il faut optimiser toutes les phases du système simultanément. Une telle optimisation est impossible car la complexité de chacun des sous-problèmes est trop grande et on ne peut l'envisager qu'en réduisant les paramètres de chacune des phases [106] : type de stratégie, nombre et nature des satellites. Pour notre étude, nous avons décidé de réduire l'optimisation de constellation à la phase de design afin de pouvoir intégrer tous les paramètres présents au cours de cette phase.

Cette phase définit les performances de la constellation en positionnant les satellites autour de la terre. Les variables de notre problème d'optimisation sont d'une part le nombre de satellites que l'on considère comme inconnu et d'autre part la position de chacun d'eux.

En positionnant le nombre de satellites comme inconnu pour le problème d'optimisation on pose un

obstacle supplémentaire : le nombre d'inconnues varie dynamiquement au cours de l'optimisation. Nous verrons au cours de la prochaine partie l'influence de cette donnée sur la régularité du critère et sur le choix de la méthode d'optimisation.

2.3 Les paramètres définissant un satellite

Un satellite est un corps artificiel placé en orbite autour de la terre. Il obéit donc à toutes les lois gravitationnelles qui régissent le fonctionnement de notre univers.

Ces lois ([20]) nous disent entre autres choses qu'un satellite (naturel ou artificiel) en orbite autour d'un astre décrit une ellipse contenue dans un plan appelé plan orbital.

Partant de ces connaissances, on peut définir le comportement d'un satellite par la donnée de 6 paramètres appelés paramètres orbitaux. Ces paramètres permettent de connaître la position du satellite à tout instant.

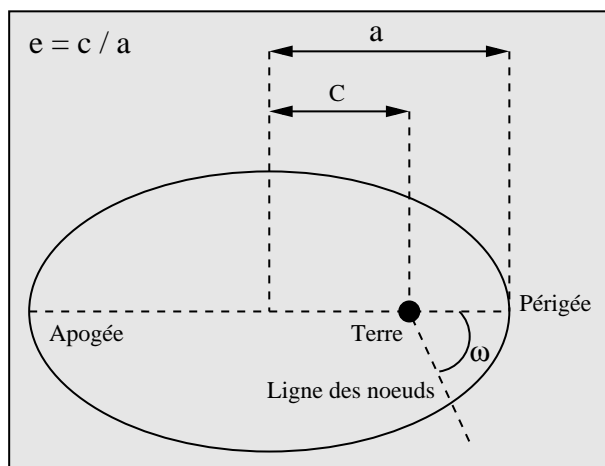


FIG. 2.1 – *Orbite elliptique*

Les deux premiers paramètres définissent la forme de l'ellipse (figure 2.1) décrite par le satellite.

- a : demi grand axe de l'ellipse.
- e : excentricité de l'ellipse.

La terre est située à l'un des deux foyer de l'ellipse et il existe deux positions remarquables sur cette orbite :

- L'apogée : c'est le point le plus éloigné de la terre
- Le périhélie : c'est le point le plus proche de la terre

Lorsque l'excentricité est nulle ($e = 0$) on parle d'orbite circulaire et la notion de périhélie et d'apogée disparaît (a devient le rayon du cercle de l'orbite).

Le paramètre suivant définit la position du satellite sur son orbite. On parle d'anomalie moyenne, notée M . C'est l'angle que ferait un satellite situé sur une orbite circulaire de même demi grand axe. Bien que ne représentant pas une donnée physique de l'orbite du satellite, cette anomalie est utilisée car on peut donner facilement sa loi horaire. D'autres anomalies existent (anomalie réelle v et excentrée E) et nous détaillerons les liens qui existent entre elles dans le chapitre 3 consacré aux critères.

Le dernier paramètre présent dans le plan orbital définit la position de l'ellipse dans ce

même plan. L'orbite peut en effet prendre n'importe quelle position du moment que la terre reste l'un des foyers. Le plan orbital passe par le centre de la terre et coupe le plan équatorial selon une droite appelée ligne des noeuds. On définit ω , appelé argument du périégée, qui n'est autre que l'angle entre la ligne des noeuds et le grand axe de l'ellipse.

Enfin, les deux derniers paramètres définissent la position du plan orbital par rapport à la terre.

- i est l'inclinaison du plan orbital par rapport au plan équatorial.
- Ω est l'angle (dans le plan équatorial) formé par la ligne des noeuds et le méridien de Greenwich : on parle de longitude du noeud ascendant.

Un satellite est donc défini par le 6-uplet de paramètres orbitaux $(a, e, i, \omega, \Omega, M)$.

Pour des raisons de simplicité d'écriture et pour se rendre compte plus facilement de la nature de l'orbite (orbite basse, moyenne, haute, ...) nous allons remplacer le paramètre a par l'altitude h du satellite au niveau du périégée lorsque nous donnerons des valeurs numériques pour le jeu de paramètres orbitaux. On a la relation suivante : $h = a(1 - e) - Re$, Re étant le rayon équatorial de la terre.

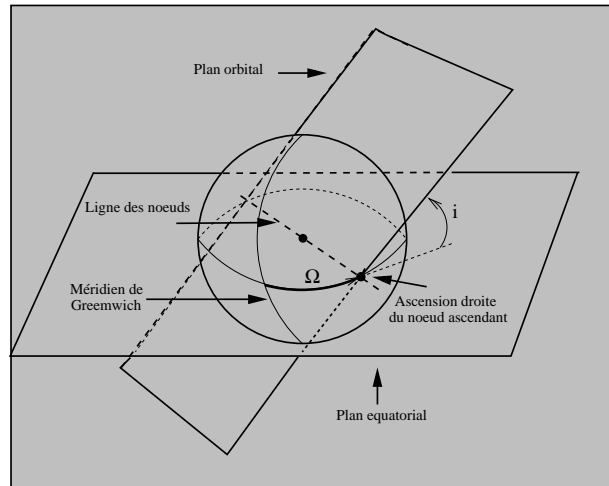


FIG. 2.2 – *Position du plan orbital*

2.4 Nature physique des paramètres

Comme nous l'avons décrit, les 6 paramètres orbitaux sont chacun rattachés à une grandeur physique différente dont les unités et domaines de variation sont :

- $h \in [400, 36000]$ exprimé en kilomètres.
- $e \in [0, 1[$ sans unité
- $i \in [0, 360]$ exprimé en degrés.
- $\omega \in [0, 360]$ exprimé en degrés.
- $\Omega \in [0, 360]$ exprimé en degrés.
- $M \in [0, 360]$ exprimé en degrés.

Cette hétérogénéité de la nature physique des paramètres (au niveau des unités et des tailles de domaine de variation) impose une mise à l'échelle lors de l'application de toute méthode d'optimisation.

Premièrement il faut noter que les variables représentant des angles sont des variables cycliques : autrement dit la borne inférieure de l'intervalle est la même (modulo 360) que la borne supérieure. Ceci doit être pris en compte lors de la recherche pour juger, par exemple, de la proximité ou de

l'éloignement de deux solutions.

Deuxièmement, les domaines de variation n'ont pas la même échelle. Nous avons décidé, pour pallier ce problème, d'homogénéiser les plages de variation en les ramenant à l'intervalle $[0,1]$. On optimise donc $(h,e,i,\omega,\Omega,M) \in [0,1]^6$.

Troisièmement, l'homogénéisation ne résout pas tous les problèmes puisque chacun des paramètres n'a pas la même influence sur le critère d'évaluation. Ceci implique qu'une modification (de même échelle) sur deux paramètres différents n'aura pas le même effet. Si l'effet sur les caractéristiques de l'orbite sont prévisibles (cf 2.5) du fait de la liaison entre le paramètre mathématique et le paramètre physique, l'effet sur le critère d'évaluation n'est pas aussi simple à apprécier.

2.5 Influence des paramètres sur l'orbite

Nous allons successivement analyser l'influence de chacun des 6 paramètres orbitaux sur l'orbite.

2.5.1 Influence de a

Le demi grand axe a est directement lié à la durée de révolution du satellite autour de son orbite et on a l'équation ([20])

$$T = 2\pi \sqrt{\frac{a^3}{\mu}} \quad (2.1)$$

μ étant la constante de gravitation.

Une augmentation de a entraîne donc une augmentation de la période de révolution du satellite sur son orbite. Pour une altitude de $400km$ ($a = 6778km$ pour une orbite circulaire) la durée de révolution du satellite est $T = 1H30mn$ et lorsque a tend vers l'altitude géostationnaire, T tend vers $24H$.

2.5.2 Influence de e

L'excentricité n'a pas d'influence sur la période de révolution du satellite mais si pour une excentricité nulle le satellite a une vitesse constante par rapport à un point situé sur la terre, ce n'est plus le cas pour une orbite elliptique ($e \neq 0$). Lorsque le satellite est proche de l'apogée, il a une vitesse (relative à la terre) beaucoup plus grande que lorsqu'il est proche de l'apogée.

2.5.3 Influence de i

La valeur de l'inclinaison permet de calculer les latitudes que le satellite peut survoler. Plus l'inclinaison augmente plus la largeur de la bande de latitude accessible augmente. C'est le déplacement nord-sud du satellite qui est affecté.

2.5.4 Influence de ω

Ce paramètre n'a d'influence que si l'excentricité est non nulle. Il permet de tirer partie de l'influence de l'excentricité sur le parcours de l'orbite. On peut par exemple positionner le passage du satellite par son apogée lorsqu'il survole une zone d'intérêt, ainsi le temps de passage au dessus de cette zone sera augmenté. C'est le déplacement est-ouest du satellite qui est affecté.

2.5.5 Influence de Ω

Si les paramètres précédents ont une influence sur le comportement du satellite au sein de son orbite, le paramètre Ω ne permet que de réaliser une rotation de celle-ci autour de la terre. On peut, en agissant sur Ω , ajuster le survol d'une zone en particulier.

2.5.6 Influence de M

L'influence de M sur le comportement du satellite est du même ordre que celle de Ω puisque l'on fixe ici la position du satellite sur son orbite (On ajuste comme avec le paramètre Ω le survol d'une zone par le satellite).

2.5.7 Influence sur les performances de la constellation

L'influence, sur les performances de la constellation, d'une variation sur un paramètre d'un satellite indépendamment des autres ne peut pas être prédite. Une constellation n'est performante que parce que les satellites qui la composent travaillent ensemble : ils sont synchronisés pour satisfaire les performances globales.

Prenons l'exemple d'une constellation de deux satellites (Sat_1, Sat_2) (figure 2.3). Le premier a une période orbitale T_1 de 12 heures ($a_1 = 26610.223km$), et le second a une période orbitale T_2 de 6 heures ($a_2 = 16763.390km$). Les deux satellites passent au dessus d'un même point à l'instant t_0 . Au bout de $12h$, les deux satellites se retrouvent dans leur position initiale (sur l'orbite). Si on modifie la valeur de a_1 de 1 % ($a_1 = 26876.325km$), la période orbitale T_1 est maintenant de 12 heures et 11 minutes. Les satellites ne sont plus synchronisés et un décalage de 22 minutes par jour aura lieu. Ce manque d'informations sur l'influence des variations des paramètres orbitaux est encore plus vrai en ce qui concerne les paramètres de positionnement de l'orbite (Ω) et du satellite (M) autour de la terre. Une modification de ces paramètres peut avoir toutes les répercussions sur les performances de la constellation. On comprend facilement avec la figure 2.3 que tout déplacement (sur son orbite) d'un satellite à l'instant t_0 aura des répercussions sur le placement de celui-ci à l'instant $t_0 + 12H$. Dans le premier cas tout déplacement conduirait à une désynchronisation alors que dans le second une variation correctement calculée du paramètre M permettrait de synchroniser les satellites à l'instant $t_0 + 12H$.

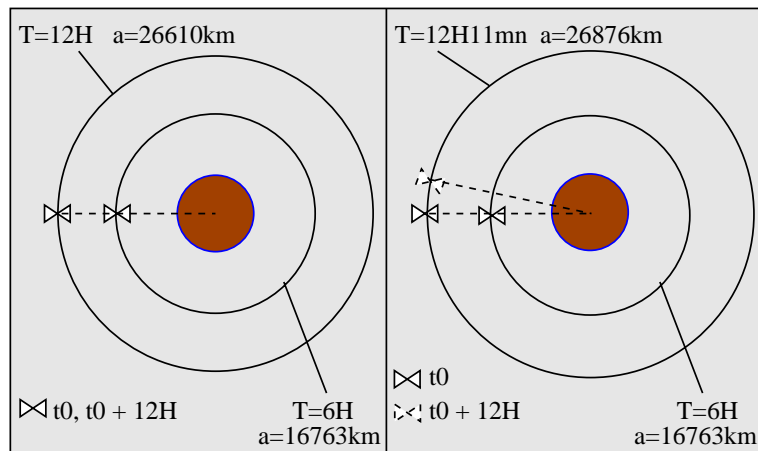


FIG. 2.3 – Exemple de désynchronisation

2.6 La nature mathématique des paramètres

Au même titre que la nature physique des paramètres orbitaux, la nature mathématique des paramètres a une influence sur la complexité du problème et sur les méthodes appropriées à sa résolution.

En effet, nous avons ici un problème en variables mixtes, c'est-à-dire que les paramètres peuvent être à valeur réelle et donc évoluer dans un intervalle continu (c'est le cas des paramètres orbitaux) ou à valeur entière donc évoluer dans un ensemble discret (c'est le cas du nombre de satellites).

Certaines études [36], proposent de résoudre le problème en fixant le nombre de satellites de la constellation avant de commencer la recherche de solutions. Cette approche permet d'appliquer des algorithmes qui ne peuvent traiter que des problèmes en variables réelles (c'est le cas des algorithmes analytiques qui nécessitent une régularité du critère que les paramètres entiers empêchent).

Dans notre cas d'étude, le nombre de satellites est supposé inconnu et utiliser une telle approche consisterait à résoudre successivement, un grand nombre de fois, le problème, en changeant le nombre de satellites de la constellation. Un tel processus d'optimisation s'approcherait de la recherche exhaustive et conduirait à des temps de calcul trop importants. Nous préférons donc intégrer dans le processus de recherche de solutions la recherche du nombre de satellites.

Conclusion

La première approche du problème fait apparaître un système intrinsèquement complexe. Celui-ci implique un grand nombre de sous-problèmes d'optimisation pour chacune des phases du système: le design de la constellation, la mise à poste et le maintien à poste. La constellation doit être optimisée au travers de ces différentes phases selon trois critères principaux: le coût, les performances et le temps.

Nous avons choisi de limiter notre champ d'étude au problème de design de constellation en intégrant comme critère les performances de la constellation.

Ce problème est un sous-problème très complexe dont les nombreux paramètres évoluent dans des domaines vastes et hétérogènes. Cette différence au niveau des paramètres se traduit par des natures physiques différentes entraînant une différence d'influence (qualitative et quantitative) selon les paramètres.

Par ailleurs, la cohabitation de variables réelles (les paramètres orbitaux) et de variables entières (le nombre inconnu n de satellites) ainsi que la taille de l'espace de recherche engendré par les $6n$ paramètres de la constellation imposent un certain nombre de contraintes quant au choix des méthodes d'optimisation à appliquer: toute méthode exigeant un taux d'exploration de l'espace important sera écartée.

Par ailleurs, nous verrons dans la partie II consacrée à l'étude des critères d'optimisation, que d'autres difficultés viennent s'ajouter à celles identifiées ici, nous contraignant, ou nous guidant, encore un peu plus dans nos choix sur les techniques à appliquer.

Partie II

Les critères

Le design de constellations a pour objectif de positionner les satellites de manière à maximiser (ou minimiser selon les cas) la valeur d'un certain critère. Cette partie va s'intéresser aux différents critères d'optimisation qui vont servir pour l'évaluation des constellations. Ces différents critères dépendent du domaine d'application de la constellation (observation, télécommunications, navigation) et des performances attendues.

Nous allons découvrir au cours de cette partie les différentes techniques intrinsèques à l'évaluation d'une constellation. En effet, quel que soit le domaine d'application de la constellation, le principe d'évaluation reste le même : simuler le comportement de la constellation pendant un certain temps et évaluer ses performances pendant cette simulation. L'étude de la simulation sera l'occasion de constater que l'évaluation d'une constellation est un processus complexe, coûteux en temps de calcul.

Au cours de cette partie, nous détaillerons successivement : les grands principes de l'évaluation d'une constellation, avec notamment le mécanisme de propagation, le calcul de visibilité des satellites et les post-traitements appliqués aux données; Ensuite nous détaillerons le calcul des performances locales des constellations dans le cadre des télécommunications et de l'observation (chapitre 4, page 65) puis de la navigation (chapitre 5, page 69).

Chapitre 3

Les grands principes de l'évaluation

3.1 La propagation des satellites

3.1.1 Introduction

La fonction d'évaluation d'une constellation de n satellites ne consiste pas seulement en l'évaluation d'une relation donnant la valeur de celle-ci en fonction des $6n$ paramètres orbitaux de la constellation.

Comme nous l'avons déjà indiqué au chapitre 2 (page 41) un satellite est un élément dynamique. Les satellites étant soumis aux lois gravitationnelles [20], ils sont animés d'un mouvement de rotation autour de la terre. L'évaluation d'une constellation consiste à simuler son évolution au cours du temps. Au niveau des satellites, on parle de propagation.

Nous nous intéressons ici au mécanisme de propagation ainsi qu'aux équations donnant la position des satellites dans les différents repères. Les calculs de performance seront détaillés dans les chapitres 4 et 5.

3.1.2 Mécanisme

Le mouvement d'un satellite autour de la terre est lié à deux phénomènes. D'une part, le mouvement rotatif de la terre entraîne le satellite, on parle de **dérive standard** des paramètres orbitaux. D'autre part, la terre n'étant pas une sphère parfaite (aplatissement au niveau de pôles), des dérives supplémentaires apparaissent. Ces dérives sont de deuxième ordre (J_2) (voir de troisième ordre (J_3)). Elles sont de moindre importance par rapport à la dérive standard et on peut raisonnablement les négliger si on considère une simulation sur une période courte. Par contre, dès qu'il s'agit de simuler l'évolution de constellations sur des périodes plus longues, leur intégration est nécessaire. Le tableau 3.1 donne l'expression des dérives standards et d'ordre 2 (J_2).

$\frac{d.}{dt}$	dérive standard	dérive liée au J_2
a	0	0
e	0	0
i	0	0
ω	0	$-\frac{3}{4}\left(\frac{R_e}{a}\right)^2 \frac{J_2}{(1-e^2)^2} \sqrt{\frac{\mu}{a^3}} (1 - 5 \cos^2 i)$
Ω	0	$-\frac{3}{2}\left(\frac{R_e}{a}\right)^2 \frac{J_2}{(1-e^2)^2} \sqrt{\frac{\mu}{a^3}} \cos^2 i$
M	$\left(\frac{\mu}{a^3}\right)^{1/2}$	$-\frac{3}{4}\left(\frac{R_e}{a}\right)^2 \frac{J_2}{(1-e^2)^{3/2}} \sqrt{\frac{\mu}{a^3}} (1 - 3 \cos^2 i)$

TAB. 3.1 – Dérives

R_e est le rayon équatorial de la terre ($R_e = 6378140$ km), μ la constante de gravitation ($\mu = 398600.47.10^9$), J_2 le facteur de dérive lié à l'aplatissement de la terre ($J_2 = 1.08262583.10^{-3}$).

Propager un satellite à un instant t revient à calculer la valeur des paramètres orbitaux à cet instant en tenant compte des différentes dérivées.

$$\begin{cases} a = a_0 + \frac{da}{dt} \cdot \Delta t \\ e = e_0 + \frac{de}{dt} \cdot \Delta t \\ i = i_0 + \frac{di}{dt} \cdot \Delta t \\ \omega = \omega_0 + \frac{d\omega}{dt} \cdot \Delta t \\ \Omega = \Omega_0 + \frac{d\Omega}{dt} \cdot \Delta t \\ M = M_0 + \frac{dM}{dt} \cdot \Delta t \end{cases}$$

$(a_0, e_0, i_0, \omega_0, \Omega_0, M_0)$ valeur des paramètres orbitaux à l'instant de référence t_0 . $\Delta t = t - t_0$

Il est intéressant de noter que seuls les paramètres indépendants du temps (a, e, i) interviennent dans les équations des dérivées des autres paramètres (ω, Ω, M). On peut donc calculer la valeur des paramètres orbitaux à n'importe quelle date si on connaît leur valeur à un instant de référence.

3.2 Les systèmes de coordonnées

Pour pouvoir évaluer les performances d'un satellite, nous avons besoin d'exprimer ses coordonnées dans un repère orthonormé centré sur la terre. Les seuls paramètres dont nous disposons sont les paramètres orbitaux ($a, e, i, \omega, \Omega, M$) du satellite. Nous allons donc successivement passer de cette représentation orbitale à l'expression des coordonnées cartésiennes du satellite dans un repère que

nous définirons.

Les coordonnées cartésiennes du satellite pourront être données arbitrairement dans l'un des deux repères suivants.

3.2.1 Le repère géocentrique (R_{geo})

C'est un repère lié à la terre dont le centre O est le centre de la terre, dont le plan (oxz) contient le méridien 0 et dont le plan (oxy) est le plan de l'équateur.

Le système de coordonnées est (X,Y,Z) pour un point quelconque de l'espace. Mais on adopte généralement le système de coordonnées sphériques (r,θ,ϕ) pour des points au voisinage de la terre. Les termes employés sont longitude à la place de θ , latitude à la place de ϕ et altitude à la place de $r - Re$.

On note R_{geo} ce repère.

Passage des coordonnées cartésiennes aux coordonnées sphériques

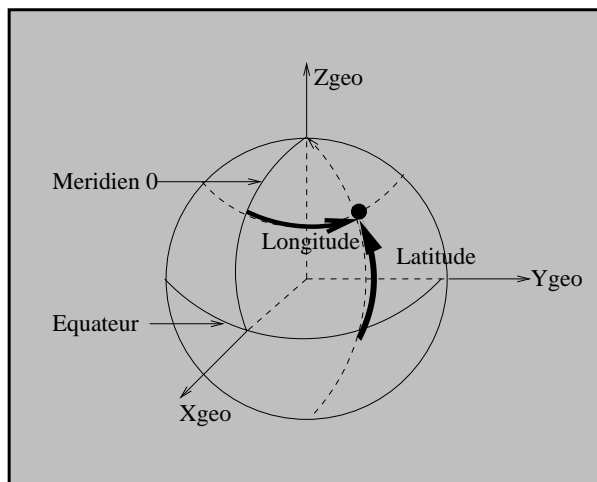


FIG. 3.1 – Repère géocentrique R_{geo}

$$x = r \cos \theta \cos \phi \quad (3.1)$$

$$y = r \sin \theta \cos \phi \quad (3.2)$$

$$z = r \sin \phi \quad (3.3)$$

Passage des coordonnées sphériques aux coordonnées cartésiennes

$$r = \sqrt{x^2 + y^2 + z^2} \quad (3.4)$$

$$\phi = \begin{cases} \text{Arctan}\left(\frac{z}{\sqrt{x^2+y^2}}\right) & \text{pour } x^2 + y^2 \neq 0 \\ \frac{\pi}{2} & \text{pour } x = y = 0, z > 0 \\ -\frac{\pi}{2} & \text{pour } x = y = 0, z < 0 \\ \text{indetermin} & \text{pour } x = y = z = 0 \end{cases} \quad (3.5)$$

$$\theta = \begin{cases} \text{Arctan}\left(\frac{y}{x}\right) & \text{pour } x > 0, y > 0 \\ \pi + \text{Arctan}\left(\frac{y}{x}\right) & \text{pour } x < 0 \\ 2\pi + \text{Arctan}\left(\frac{y}{x}\right) & \text{pour } x > 0, y < 0 \\ \frac{\pi}{2} & \text{pour } x = 0, y > 0 \\ 3\frac{\pi}{2} & \text{pour } x = 0, y < 0 \\ \text{indetermin} & \text{pour } x = y = 0 \end{cases} \quad (3.6)$$

3.2.2 Le repère galiléen (R_{gal})

C'est un repère fixe dans lequel la terre a un mouvement de rotation d'axe (oz). On note R_{gal} ce repère.

Il coïncide avec le repère géocentrique à l'instant 0.

On note w_T l'angle de rotation entre le plan (oxz) de R_{gal} et le plan (oxz) de R_{geo} . Le système de coordonnées est (X,Y,Z).

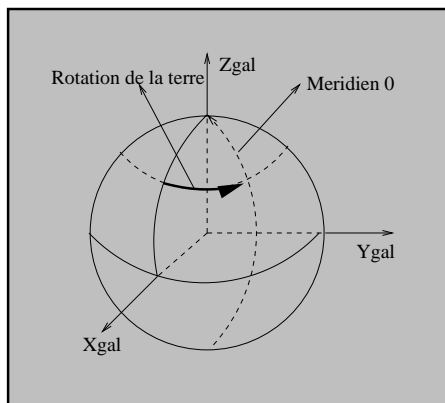


FIG. 3.2 – Repère galiléen R_{gal}

3.3 Équations de mouvement du satellite

3.3.1 Équation dans le plan orbital

Nous allons nous situer dans le plan orbital (figure 3.3) pour donner la première expression des équations de mouvement du satellite.

De manière générale, le satellite décrit une ellipse dans le plan orbital dont la terre est l'un des foyers, dont l'excentricité est e et dont le demi-grand axe est a ([20]). Le mouvement du satellite dans ce repère va être calculé à partir de l'expression de l'anomalie moyenne $M(t)$.

On définit par ailleurs trois types d'anomalie.

1. $v(t)$ l'anomalie réelle, angle rapporté au foyer de l'ellipse (terre).
2. $E(t)$ l'anomalie excentrée, angle rapporté au centre de l'ellipse.
3. $M(t)$ l'anomalie moyenne, angle rapporté à une orbite circulaire de même période que l'orbite elliptique.

On a les relations suivantes entre les différentes anomalies :

$$\cos E(t) = \frac{\cos v(t) + e}{1 + e \cos v(t)} \quad (3.7)$$

$$M(t) = E(t) - e \sin E(t) = n(t - t_0) \quad (3.8)$$

n étant le mouvement moyen du satellite sur l'orbite.

$$n = \frac{2\pi}{T} = \sqrt{\frac{\mu}{a^3}} \quad (3.9)$$

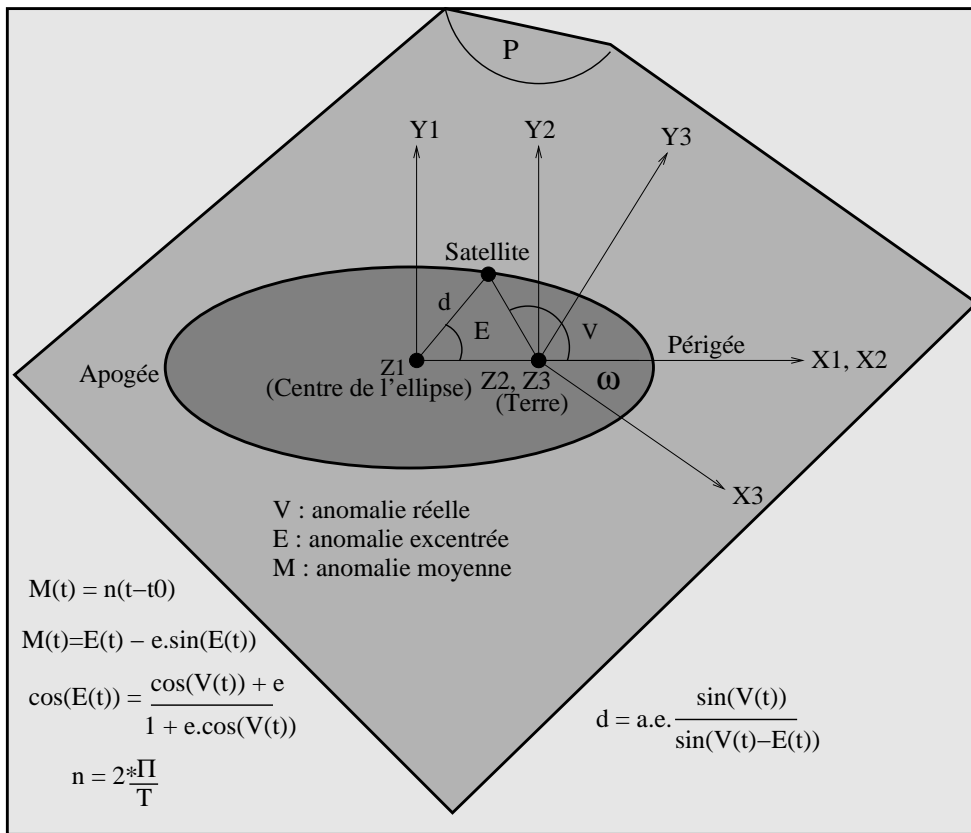


FIG. 3.3 – *Le plan orbital*

On connaît donc la loi horaire de M mais il nous faut déterminer celle de E pour pouvoir utiliser l'équation polaire de l'ellipse et ainsi avoir l'expression, dans le repère lié à l'ellipse, du mouvement du satellite. L'équation 3.8 peut être résolue par approximations itératives avec la méthode Newton ([34]). La période de révolution autour de l'orbite est donnée par:

$$T = 2\pi \sqrt{\frac{a^3}{\mu}} \quad (3.10)$$

On définit R_1 comme étant le repère lié à l'ellipse, c'est-à-dire de centre celui de l'ellipse, tel que l'axe (ox_1) soit le grand axe de l'ellipse et que l'axe (oy_1) soit le petit axe de l'ellipse (cf figure 3.3). Les équations polaires de l'ellipse donnent l'équation de mouvement dans R_1 (cf figure 3.3).

$$(x_1(t), y_1(t), z_1(t)) = (d \cos E(t), d \sin E(t), 0) \quad (3.11)$$

avec $b = a\sqrt{1 - e^2}$.

On définit ensuite le repère R_2 déduit du repère R_1 par une translation de vecteur $(e * a, 0, 0)$ (cf 3.3).

L'équation du satellite dans R_2 est donc

$$(x_2(t), y_2(t), z_2(t)) = (x_1(t) - ea, y_1(t), z_1(t)) \quad (3.12)$$

$$(x_2(t), y_2(t), z_2(t)) = (d \cos E(t) - ea, d \sin E(t), 0) \quad (3.13)$$

On intègre ensuite la donnée de l'argument du périégée ω pour définir le repère R_3 obtenu à partir de R_2 par rotation d'angle $-\omega$ et d'axe $(o2, z2)$ (cf 3.3).

L'équation du satellite dans R_3 est

$$(x_3(t), y_3(t), z_3(t)) = Rot_3.(x_2(t), y_2(t), z_2(t))^t \quad (3.14)$$

avec Rot_3 la matrice de rotation d'angle $-\omega(t)$ et d'axe (oz_2) et dont l'expression est

$$\begin{bmatrix} \cos(-\omega(t)) = \cos \omega(t) & \cos(-\omega(t) + \frac{\pi}{2}) = \sin \omega(t) & 0 \\ \cos(-\omega(t) - \frac{\pi}{2}) = -\sin \omega(t) & \cos(-\omega(t)) = \cos \omega(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

3.3.2 Équation dans le repère géocentrique

On déduit ensuite le repère R_4 du repère R_3 par une rotation d'angle $-i(t)$ et d'axe (ox_3) .

$$(x_4(t), y_4(t), z_4(t)) = Rot_4.(x_3(t), y_3(t), z_3(t))^t \quad (3.16)$$

avec Rot_4 la matrice de rotation d'angle $-i(t)$ et d'axe (ox_3)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-i(t)) = \cos i(t) & \cos(-i(t) + \frac{\pi}{2}) = \sin i(t) \\ 0 & \cos(-i(t) - \frac{\pi}{2}) = -\sin i(t) & \cos(-i(t)) = \cos i(t) \end{bmatrix} \quad (3.17)$$

Enfin l'expression du mouvement du satellite dans le repère géocentrique R_{geo} est donnée par :

$$(x_{geo}(t), y_{geo}(t), z_{geo}(t)) = Rot_{geo}.(x_4(t), y_4(t), z_4(t))^t \quad (3.18)$$

avec Rot_{geo} la matrice de rotation d'angle $-\Omega(t)$ et d'axe (oz_4) dont l'expression est

$$\begin{bmatrix} \cos(-\Omega(t)) = \cos \Omega(t) & \cos(-\Omega(t) + \frac{\pi}{2}) = \sin \Omega(t) & 0 \\ \cos(-\Omega(t) - \frac{\pi}{2}) = -\sin \Omega(t) & \cos(-\Omega(t)) = \cos \Omega(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

On peut également donner l'expression des coordonnées du satellite dans le repère galiléen R_{gal} .

$$(x_{gal}(t), y_{gal}(t), z_{gal}(t)) = Rot_{gal} \cdot (x_{geo}(t), y_{geo}(t), z_{geo}(t))^T \quad (3.20)$$

avec Rot_{gal} la matrice de rotation d'angle $-w_T(t)$ (angle de rotation de la terre) et d'axe (oz_{geo}) dont l'expression est

$$\begin{bmatrix} \cos(-w_T(t)) = \cos w_T(t) & \cos(-w_T(t) + \frac{\pi}{2}) = \sin w_T(t) & 0 \\ \cos(-w_T(t) - \frac{\pi}{2}) = -\sin w_T(t) & \cos(-w_T(t)) = \cos w_T(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

3.4 La simulation

La simulation consiste à faire évoluer la constellation et à calculer ses performances au cours du temps. Cette évaluation nécessite un échantillonnage du temps et de l'espace (paragraphe 3.4.1) pour permettre une évaluation des performances locales de la constellation (chapitre 4 et 5). Par ailleurs la constellation est simulée durant une certaine période de temps dont le principe de calcul est donné dans le paragraphe 3.4.3.

3.4.1 Échantillonnage du temps et de l'espace

Pour les besoins de la simulation, on est amené à discretiser à la fois les zones d'intérêt que l'on considère et la période de simulation de la constellation. Cette étape, bien que secondaire, peut permettre de réduire les temps de calcul sous certaines hypothèses. En effet, le pas d'échantillonnage en temps et en espace dépend des paramètres de la constellation. Par exemple pour une constellation en orbite basse le défilement des satellites est très rapide, et un pas de quelques minutes peut suffire pour laisser passer des trous de visibilité lors de la simulation.

Il en est de même pour l'échantillonnage de l'espace. Le pas dépend des paramètres du satellite (altitude, inclinaison) et peut ne pas être uniforme sur toutes les zones (cas de zones prioritaires) (figure 3.4).

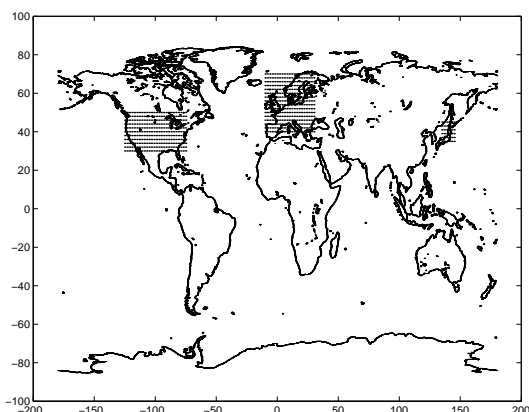


FIG. 3.4 – Échantillonnage de l'espace

3.4.2 Notion de trace au sol

La droite passant par le satellite et par le centre de la terre traverse la surface terrestre en un point appelé **point sous satellite** (figure 3.5). Si on se réfère à l'expression de la position du satellite en coordonnées sphériques dans le repère géocentrique (cf 3.2.1, page 53) qui est donnée sous la forme $(long_s, lat_s, alt_s)$, le point sous satellite a pour coordonnées sphériques (lon_s, lat_s, Re) . On définit la **trace** d'un satellite comme étant l'ensemble des points sous satellite obtenus lors de la propagation du satellite sur son orbite. Un point particulier de la trace est le point sous-satellite qui correspond au passage du satellite de l'hémisphère sud à l'hémisphère nord. C'est l'endroit où la trace coupe l'équateur, et si les paramètres orbitaux sont donnés à la date à laquelle le satellite passe par ce point alors sa longitude n'est autre que la longitude du noeud ascendant (paramètre orbital Ω).

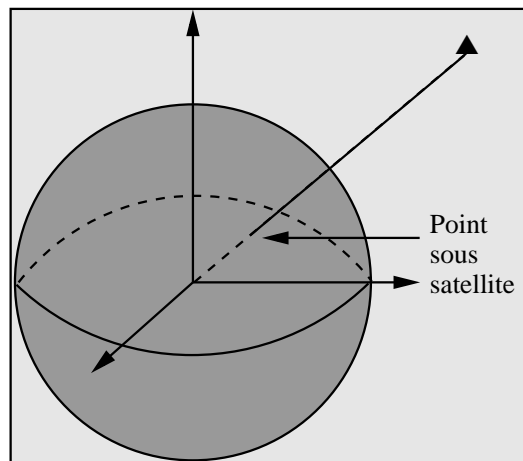


FIG. 3.5 – *Point sous satellite*

En fonction de la valeur des paramètres orbitaux du satellite, la forme de la trace au sol peut être très différente. La trace d'un satellite géostationnaire est un unique point (figure 3.6), la trace d'un satellite d'altitude géostationnaire mais ayant une inclinaison non nulle forme une sorte de 8 (figure 3.7), plus communément la trace d'un satellite en orbite basse ressemble à une sinusoïde (figure 3.8). Cette trace peut également se répéter après un certain nombre de révolutions du satellite autour de son orbite. C'est par exemple le cas des satellites de la constellation *GPS* qui répètent leur trace au bout de 12h (figure 3.9).

Cette différence de trace permet déjà d'entrevoir des différences au niveau des services que peuvent rendre les satellites. Ces différences seront exploitées pour constituer une base de connaissances sur les orbites (chapitre 11, page 133).

3.4.3 La durée de simulation

L'évaluation d'une constellation est donc basée sur une simulation de l'évolution de celle-ci au cours du temps. Pour être exploitable, la simulation doit être bornée dans le temps, mais pour rendre compte des performances de la constellation durant toute sa phase opérationnelle il faut être sûr que la simulation est significative. Pour limiter la durée de simulation, tout en étant sûr de ne pas passer à côté de baisses de performance de la constellation, on utilise généralement des **orbites à répétition de trace**.

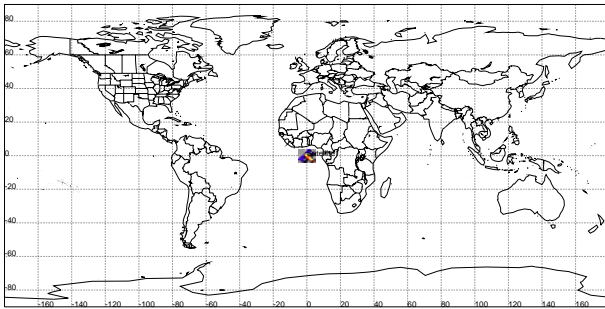


FIG. 3.6 – *Satellite GEO*

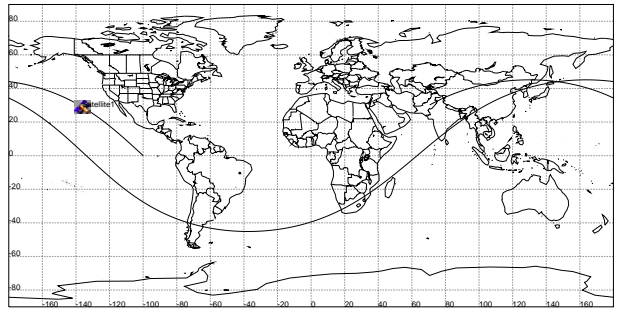


FIG. 3.8 – *Satellite LEO*

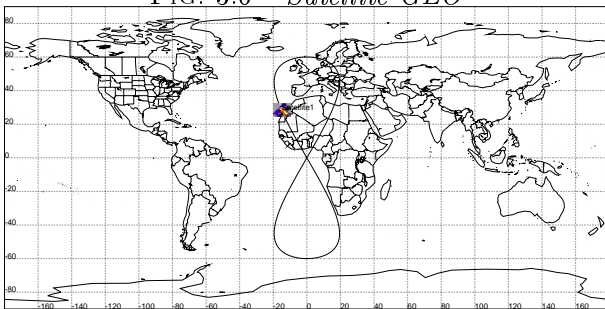


FIG. 3.7 – *Satellite IGSO*

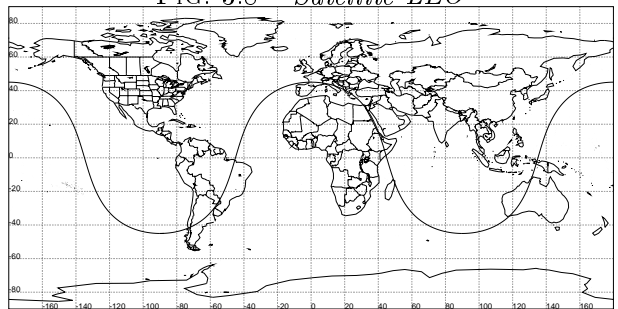


FIG. 3.9 – *Satellite GPS*

Une orbite est dite à répétition de trace lorsque la trace du satellite repasse exactement sur elle-même (figure 3.11). Si la terre n'était pas animée d'un mouvement de rotation, la période de répétitivité serait égale à une période de révolution du satellite sur son orbite. Mais la rotation de la terre entraîne un décalage de la trace à chaque révolution (figure 3.10).

La propriété de répétition de trace peut sembler contraignante car elle limite les valeurs possibles des paramètres orbitaux (a, e, i, ω) mais les domaines de variation étant suffisamment vastes, le nombre de solutions potentielles reste très grand. D'autre part, la répétition de trace peut résoudre certains problèmes lors de la phase de design ou de contrôle des satellites. Si un satellite repasse au-dessus d'une même station tous les jours à la même heure, cela facilite la mise en place des contrôles. De la même manière dans le domaine des télécommunications, les demandes sont souvent concentrées sur certaines plages horaires et la synchronisation du passage d'un satellite avec ces plages permet d'avoir de bonnes performances ([63]).

Par ailleurs, plus le nombre de révolutions avant répétition est faible, plus la durée de la simulation sera courte, mais les domaines de variation des paramètres orbitaux, seront eux plus restreints.

Il faut également noter que la période de répétitivité de la constellation peut être calculée en fonction de celles des satellites qui la composent. Par exemple si on considère une constellation de 3 satellites (S_1, S_2, S_3) ayant comme périodes de répétitivités respectives $12h, 24h$, et $36h$ celle de

la constellation sera de $72h$. En outre, sous certaines hypothèses de symétrie de la constellation, il peut exister une période de répétitivité pour la constellation qui soit plus petite que celle des satellites qui la composent. Dans ce cas de figure, c'est la trace d'un satellite i qui recouvre celle d'un satellite j .

De manière générale, on parle d'orbites phasées pour désigner de telles orbites. On dit qu'une orbite phasée réalise $p + \frac{r}{q}$ révolutions par jour. C'est-à-dire que le satellite repasse au dessus du même point après q jours et qu'il a réalisé $pq + r$ révolutions entières.

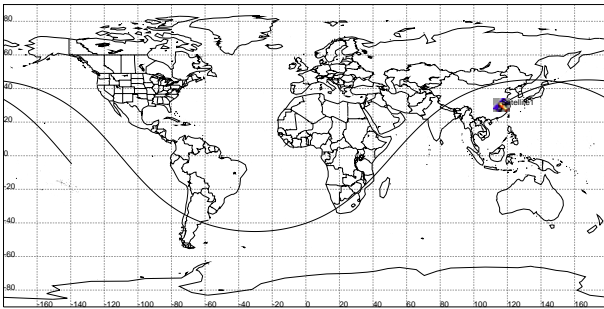


FIG. 3.10 – Non répétition de trace

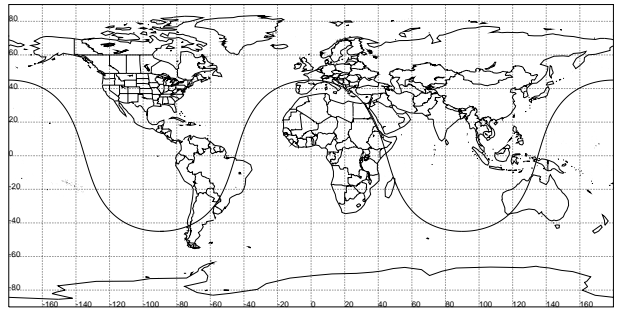


FIG. 3.11 – Répétition de trace

3.4.4 Les post-traitements

Avant de nous intéresser au calcul des performances locales de la constellation, nous allons nous intéresser au traitement des données issues de ces calculs.

En effet, si on considère un maillage de la zone d'intérêt (chaque point de maille représentant un utilisateur potentiel) en M mailles, et une discrétisation de la durée de simulation en T instants, on effectue $M * T$ calculs de performances (1 pour chaque échantillon (maille, instant)). Cet important volume de données brutes doit subir un post-traitement pour être exploitable par l'algorithme, la comparaison de deux constellations ne se faisant que sur une seule valeur (ou sur quelques valeurs dans le cas d'une optimisation multi-critère).

En fonction du domaine d'application, les post-traitements peuvent différer mais il sont souvent liés au calcul de valeurs maximales, minimales ou de moyennes sur les valeurs des échantillons obtenues au cours de la simulation. On a donc un critère final de type *min-min*.

$$\text{Min}_{t \in [1, T]} (\text{Min}_{m \in [1, M]} (f(Cst, t, m)))$$

$f(Cst, t, m)$ étant la performance de la constellation Cst à l'instant t à la maille m .

3.5 Visibilité des satellites

Après avoir déterminé la position des satellites, il faut déterminer sous quelles conditions un utilisateur situé sur la terre peut, ou non, bénéficier des services d'un satellite. Pour ce faire, on doit tout d'abord savoir si le satellite est visible depuis un point donné, situé sur la terre. Pour réaliser ce calcul, nous avons besoin de définir un certain nombre de notions comme celles de "trace au sol" ou de "cercle de visibilité". Après avoir abordé ces notions nous détaillerons les calculs de visibilité d'un point par un satellite (et inversement).

3.5.1 Calcul du cercle de visibilité

La détermination de la position des satellites par le mécanisme de propagation (cf. 3.1) et l'expression de leurs coordonnées dans un repère adéquat (cf. 3.2) sont une première étape dans le calcul des performances d'une constellation.

Il est par ailleurs souvent nécessaire de connaître, pour un point situé sur la surface terrestre, l'ensemble des satellites visibles depuis celui-ci. Le calcul de visibilité nécessite la donnée de plusieurs paramètres :

- Les coordonnées du satellite ($long_{sat}, lat_{sat}, alt_{sat}$)
- Les coordonnées du point ($long, lat, alt$).
- L'angle d'élévation minimale (ξ).

Le satellite ne sera visible depuis le point que si son élévation est supérieure à ξ . Réciproquement on est souvent amené à déterminer l'ensemble des points visibles depuis un satellite. Cet ensemble est délimité par le cercle de visibilité et les relations permettant de le déterminer sont données par la figure 3.12. On cherche à calculer le rayon angulaire θ du cercle de visibilité en fonction de l'altitude h et de l'angle d'élévation minimale ξ .

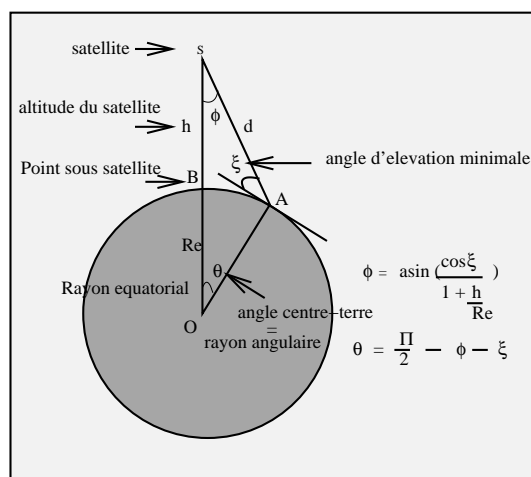


FIG. 3.12 – Cercle de visibilité

$$\Phi = \text{asin}\left(\frac{\cos \xi}{1 + \frac{h}{Re}}\right) \quad (3.22)$$

$$\theta = \frac{\pi}{2} - \phi - \xi \quad (3.23)$$

3.5.2 Bande de visibilité

3.5.3 Cas d'instruments optiques

La propagation du satellite, et donc de son cercle de visibilité, conduit à l'obtention d'une bande de visibilité. Pour savoir si un point de la surface terrestre est visible depuis le satellite, il suffit de tester l'appartenance de celui-ci au cercle de visibilité.

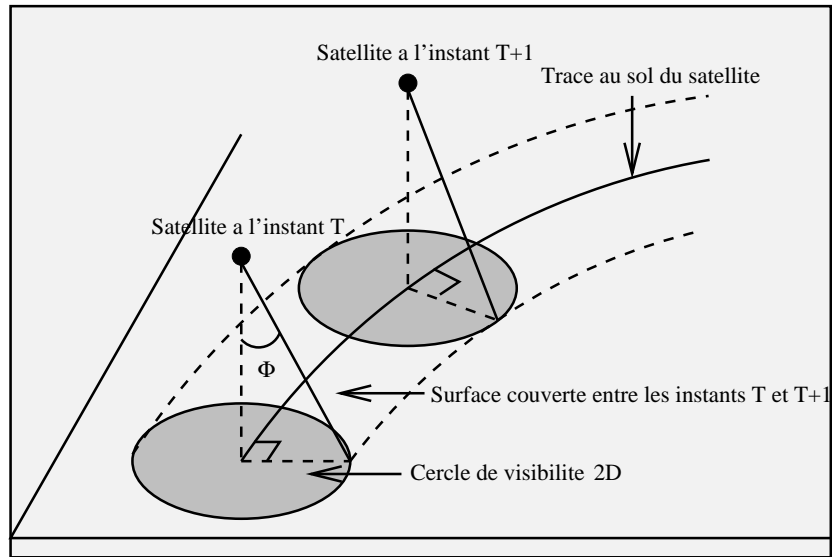


FIG. 3.13 – *Instrument optique*

3.5.4 Cas des instruments radars

Principe

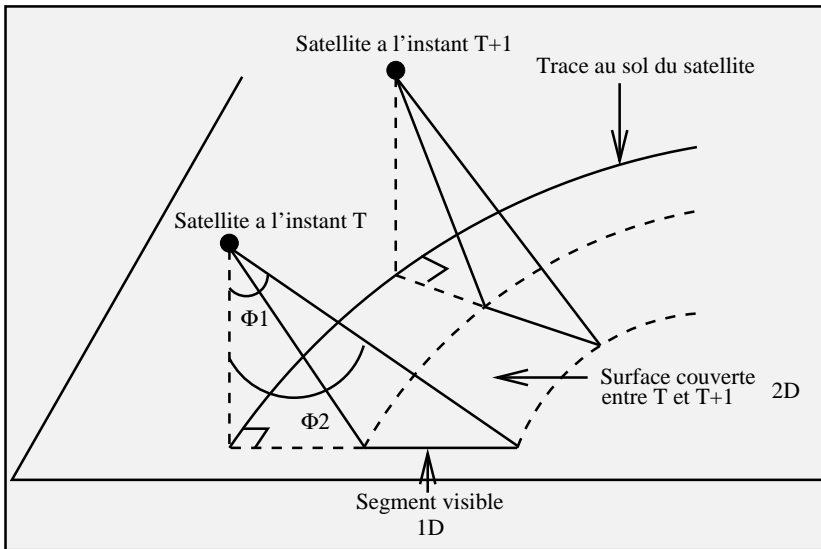


FIG. 3.14 – *Instrument radar*

Dans le cas d'instruments radars, la zone couverte à un instant t est un segment de droite (1 dimension) orthogonal au vecteur vitesse du satellite et caractérisé par les angles ϕ_1, ϕ_2 . Outre le calcul de la zone qui diffère, le test d'appartenance d'une station au segment ne peut pas être réalisé aussi simplement que dans le cas optique. En effet, la simulation étant discrétisée dans le temps, une station peut être visible au cours de celle-ci sans forcément coïncider avec un échantillon : la zone visible n'a qu'une dimension. On résout ce problème en considérant la zone balayée par l'instrument entre deux pas de temps (on retrouve ainsi les 2 dimensions).

Calcul d'appartenance

On calcule dans un premier temps, les coordonnées des points extrêmes du segment. Le principe est ensuite de tester l'appartenance du vecteur centre terre-station à l'angle solide formé par les quatre points extrêmes des segments de visibilité aux instants T et $T + 1$.

Soient $A(\lambda_A, \mu_A)$ et $B(\lambda_B, \mu_B)$ les points extrêmes du segment de visibilité à l'instant T .

Soient $C(\lambda_C, \mu_C)$ et $D(\lambda_D, \mu_D)$ les points extrêmes du segment de visibilité à l'instant $T + 1$.

Soit $M(\lambda, \mu)$ la station.

Soient :

$$S_1 = \text{signe}(\overrightarrow{MA} \wedge \overrightarrow{MB})$$

$$S_2 = \text{signe}(\overrightarrow{MB} \wedge \overrightarrow{MC})$$

$$S_3 = \text{signe}(\overrightarrow{MC} \wedge \overrightarrow{MD})$$

$$S_4 = \text{signe}(\overrightarrow{MD} \wedge \overrightarrow{MA})$$

M appartient à la surface formée par les sommets A, B, C et D si et seulement si $S_1 = S_2 = S_3 = S_4$.

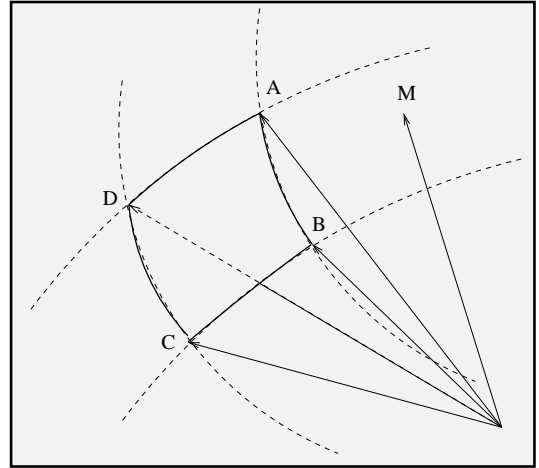


FIG. 3.15 – Angle solide

3.6 Bilan

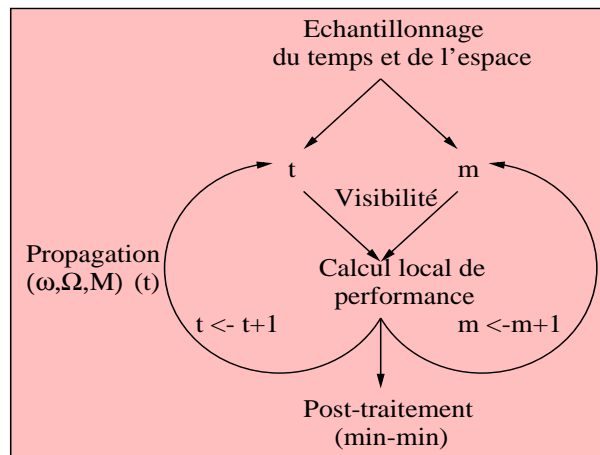


FIG. 3.16 – Bilan de l'évaluation d'une constellation

Chapitre 4

Critères dédiés aux télécommunications et à l'observation

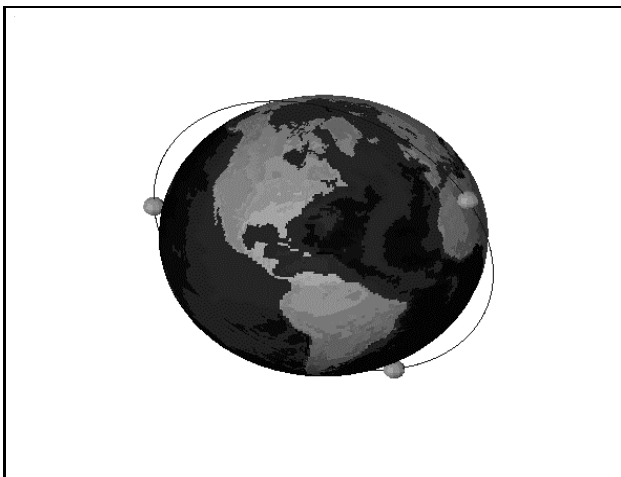


FIG. 4.1 – *Constellation Spot*

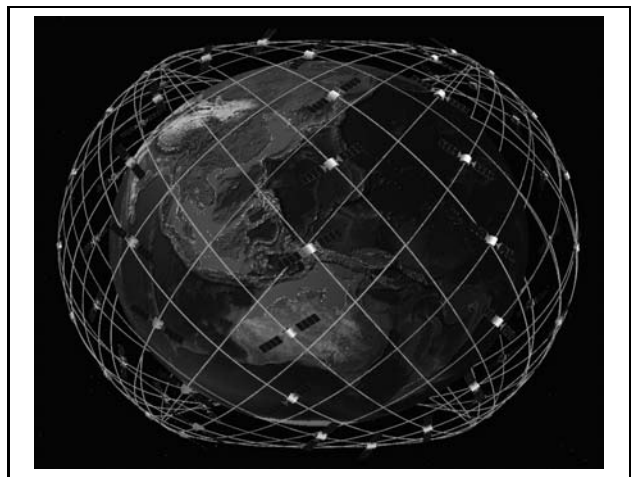


FIG. 4.2 – *Constellation Skybridge*

4.1 Introduction

Ce premier chapitre sur le détail du calcul de performance d'une constellation regroupe deux grands domaines d'application des constellations de satellite : les télécommunications et l'observation. Nous avons décidé de regrouper ces deux domaines, bien que les critères utilisés ne soient pas les mêmes car le premier ne présente pas beaucoup de particularités par rapport aux détails techniques du chapitre précédent et le second ne sera pas très détaillé puisque nous ne présenterons

de résultats dans le cadre de l'observation qu'à travers une méthode d'approche de la résolution (méthode de Hanson chapitre 11).

Le choix de ne pas appliquer notre algorithme de résolution, que nous présenterons au cours de la partie IV, aux problèmes d'observation est lié à plusieurs contraintes que nous détaillerons au cours de cette même partie.

Il faut bien noter que nous présentons un ensemble non exhaustif de critères mais qui nous semblent néanmoins représentatifs. Nous allons commencer par détailler le critère le plus simple, celui utilisé en télécommunications avant de présenter le critère utilisé en observation.

4.2 Critère dédié aux télécommunications

4.2.1 Principe

Lorsque l'on "design" une constellation destinée aux télécommunications, deux questions doivent être posées pour valider celle-ci :

- *est-ce qu'un utilisateur ayant établi une connexion pourra la conserver pendant toute la durée de la communication?* : c'est la notion de continuité temporelle du service.
- *Est-ce qu'un utilisateur ayant établi une connexion pourra la conserver lorsqu'il se déplace à la surface de la terre?*: c'est la notion de continuité spatiale du service.

Les spécifications pour un tel système vont définir les limites géographiques d'utilisation de la constellation ainsi que les limites temporelles. Si on ne conçoit pas un service de télécommunication grand public qui ne puisse fonctionner qu'à certaines heures de la journée, on peut néanmoins comprendre qu'un ensemble de satellites pourrait répondre à une augmentation ponctuelle du nombre de connexions (aux heures pleines par exemple) au dessus de zones particulières (les villes occidentales par exemple). De la même manière, des systèmes de communication par satellites dédiés à des applications scientifiques n'ont pas nécessairement besoin d'être opérationnels en permanence. Ces utilisations d'une constellation de satellites justifient la définition spatio-temporelle des performances attendues de la constellation.

Les spécifications vont par ailleurs définir le niveau de performance attendu pour la constellation en termes de taux de couverture. Nous avons vu qu'un satellite couvre à chaque instant une certaine surface terrestre et le taux de couverture en un point P à un instant t donné est le nombre de satellites de la constellation visibles depuis ce point P à l'instant t . Le taux de couverture requis pour une application dépend de la bande passante que l'on souhaite fournir aux utilisateurs ainsi que de la robustesse attendue du système. Si une constellation doit toujours fournir le service pour lequel elle a été lancée et qu'un satellite tombe en panne, il faut qu'un autre prenne le relais immédiatement. On peut être amené, pour résoudre ce problème, à définir une constellation qui assure un taux de couverture supérieur à celui nécessaire pour absorber toutes les communications. Nous

reviendrons sur la notion de panne dans le chapitre 5.

Nous poserons donc le problème de design d'une constellation de satellites dédiée aux télécommunications de la manière suivante : définir une constellation assurant un taux de couverture Tx au dessus d'une zone Z durant une période T ([30],[74]). L'évaluation du critère se résume donc à un calcul de visibilité (cf paragraphe 3.5 chapitre 3). On mémorise pour chaque instant et chaque point d'intérêt le taux de couverture fourni par la constellation.

4.2.2 Les post-traitements

Après avoir calculé un taux de couverture au dessus de chaque point d'intérêt et pendant toute la période utile, il faut calculer une valeur pour la constellation. Comme dans le cas général présenté au chapitre précédent, les post traitements sont souvent des calculs de pire cas sur la simulation mais nous lui préférons un calcul plus représentatif de la constellation en appliquant par exemple une moyenne du taux de couverture sur la simulation.

Soit $t \in D_t$ la variable temporelle évoluant dans le domaine D_t . Soit $P \in D_p$ la variable d'espace évoluant dans l'ensemble D_p des points d'intérêt. Soit $f(t,P)$ la fonction qui à un instant t et un point P calcul le taux de couverture fourni par la constellation. Soit $Tx(t,P)$ le taux minimal de couverture attendu en un point P à un instant t .

- Calcul du pire cas de taux de couverture : $Min_{t \in D_t}(Min_{P \in D_p}(f(t,P)))$
- Calcul de la moyenne du taux de couverture : $\frac{1}{Card(D_t) \times Card(D_p)} \sum_{t \in D_t} \sum_{P \in D_p} f(t,P)$
- Pourcentage de temps où le taux de couverture minimale est respecté :
 $\frac{1}{Card(D_t) \times Card(D_p)} \sum_{t \in D_t} \sum_{P \in D_p} (f(t,P) \geq Tx(t,P))$

$(f(t,P) \geq Tx(t,P))$ est une condition logique et vaut 1 si elle est vraie, 0 sinon.

D'autres post traitements peuvent être appliqués en intégrant notamment la notion de continuité temporelle du service. On calcule dans ce cadre la durée moyenne d'un service (sans discontinuité).

4.3 Critère dédié à l'observation

4.3.1 Présentation

Nous allons maintenant détailler le principe de calcul d'un critère dédié à l'optimisation des constellations d'observation. Dans le cadre de l'observation, la contrainte précédente sur la continuité du service dans le temps disparaît. Les éléments observés (forêts, océans, zones urbaines, ...) ne nécessitent pas un suivi continu et les applications basées sur de tels systèmes étant rarement commerciales, on ne peut envisager un nombre suffisant de satellites pour couvrir continûment les zones d'intérêt. Pour des problèmes d'observation, la contrainte temporelle se pose dans les termes suivants : un satellite de la constellation doit repasser au dessus d'un point quelconque au plus tard

après une période de temps Δt fixée par les spécifications. On parle de délai de revisite maximal ([75]).

En revanche, la notion de continuité spatiale reste d'actualité et l'on souhaite que sur la zone définie dans les spécifications, le service rendu soit le même quel que soit un point pris au hasard.

En conclusion, cette formulation du critère est fondamentalement différente de celle utilisée dans le cadre des télécommunications ou de la navigation puisque la notion temporelle permettant d'évaluer une performance locale de la constellation de manière décorrélée des autres instants disparaît.

4.3.2 Expression de la fonction critère

Après avoir détaillé la manière de calculer le délai de revisite d'un satellite au dessus d'un même point, nous allons donner l'expression d'une fonction critère utilisée par Hanson ([75]) pour définir une constellation minimisant le délai de revisite.

Soit D_p le domaine de variation des points d'intérêt. Soit $Rev(P)$ le délai de revisite maximal souhaité pour le point $P \in D_p$.

$$F = \sum_{P \in D_p} \{f(P) - Rev(P) + 0.6 \cdot |f(P) - Rev(P)|\} \quad (4.1)$$

Pour plus de détails sur l'origine des deux membres, et la justification des poids, on peut se reporter à l'article de Hanson [75].

Chapitre 5

Calcul du critère dans le cas de la navigation

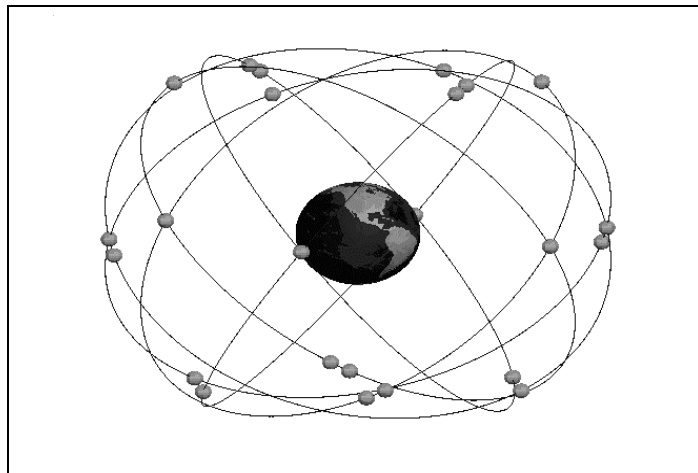


FIG. 5.1 – *Constellation GPS*

5.1 Introduction

L'optimisation de constellations dédiées à la navigation est un problème complexe car les performances de la constellation vont dépendre de la position relative des satellites les uns par rapport aux autres. Par ailleurs, pour des raisons que nous détaillerons ultérieurement, le taux de couverture minimal assuré par la constellation doit être au minimum de 4.

Le domaine de la navigation (civile ou militaire) est certainement le plus contraint dans la mesure

où les systèmes faisant appel à ce service sont souvent des systèmes critiques (système de positionnement des avions, ...).

Trois grands critères sont donc utilisés pour définir les constellations de navigation : la précision, la disponibilité et l'intégrité. Le critère de précision représente la partie technique de l'évaluation de la constellation. On attend de la constellation qu'elle permette à un utilisateur de déterminer sa position avec une certaine précision. La disponibilité ([80], [106]) représente le pourcentage de temps où le système fournit le service nominal attendu. Ce critère prend notamment en compte la tolérance de la constellation face aux pannes des satellites. Si une constellation d'observation de la terre tombe en panne, les pertes ne sont que des images; si une constellation de télécommunications n'assure plus son service, la seule conséquence sera une saturation des autres canaux de communication. Par contre, si une constellation de navigation n'est plus disponible, les systèmes embarqués dans les avions perdent une information vitale pour le repérage, ce qui peut causer de graves désagréments surtout lors des phases d'atterrissage et de décollage. Enfin, l'intégrité du système ([16]) est un critère très important car il valide ou invalide les données fournies par la constellation mais il n'intervient pas dans la phase de design de la constellation.

5.2 Principe de triangulation

Avant d'étudier en détail le calcul du critère, nous allons présenter le principe du calcul de la position d'un utilisateur (cf. modélisation mathématique des critères [54]).

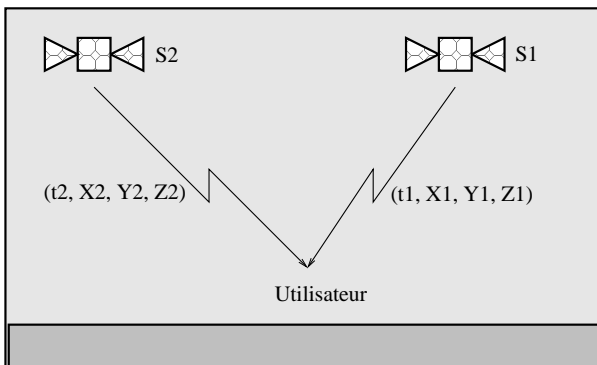


FIG. 5.2 – Émission d'un signal

On suppose que n satellites, de vecteurs positions respectifs $R_i = (X_i, Y_i, Z_i)$, $i \in [1, n]$, sont visibles par un utilisateur de vecteur position $r_u = (X, Y, Z)$. Les n satellites envoient régulièrement leur position ainsi que la date de l'envoi $(t_i)_{i \in [1, n]}$ du signal (cf. figure 5.2) (date mesurée avec l'horloge $GPS H_{gps}$). L'utilisateur reçoit le signal au temps tu_i mesuré avec sa propre horloge (H_u).

L'horloge des satellites n'étant pas synchronisée avec celle de l'utilisateur il existe un biais τ , inconnu, entre les deux horloges tel que

$$H_u = H_{gps} + \tau \quad (5.1)$$

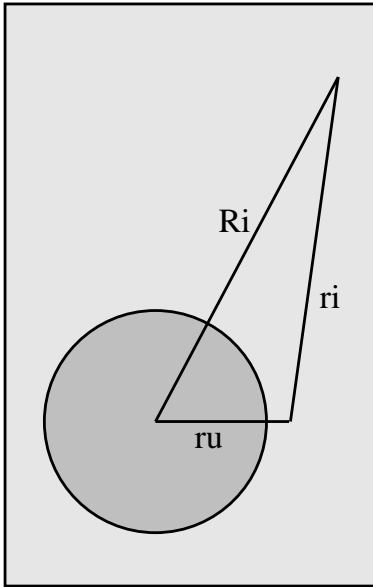


FIG. 5.3 – *Positionnement*

Soit r_i la distance réelle entre l'utilisateur et le i^{eme} satellite de la constellation GPS. On a la relation suivante

$$r_i = c((t_i - \tau) - tu_i) \quad (5.2)$$

c étant la vitesse de propagation du signal.

On appelle pseudo-distance, notée r'_i , la distance mesurée par l'utilisateur. Cette distance diffère de la distance réelle par le biais d'horloge.

$$r'_i = c(t_i - tu_i) \quad (5.3)$$

d'après les équations 5.1, 5.2, 5.3, on obtient la relation 5.4.

$$r_i = r'_i - c * \tau \quad (5.4)$$

par ailleurs (figure 5.3),

$$r_i = \|R_i - r_u\| \quad (5.5)$$

D'après l'équation 5.3 on en déduit que

$$r'_i = r_i + c\tau = \sqrt{(X - X_i)^2 + (Y - Y_i)^2 + (Z - Z_i)^2} + c\tau \quad (5.6)$$

A partir des n équations sur les n satellites en vue, on peut calculer les quatre inconnues du problème (X, Y, Z, τ) . De manière géométrique, cette résolution consiste à calculer l'intersection de n sphères centrées sur les n satellites et dont le rayon est évalué par l'utilisateur (cf. figure 5.4). Deux problèmes se posent alors

- Les équations ne sont pas linéaires et on ne peut résoudre analytiquement le problème.
- Si n est supérieur à quatre ($n \geq 4$), on a un système sur-déterminé.

Pour résoudre le problème, on utilise une estimation aux moindres carrés, dont les mesures sont les pseudo-distances calculées par l'utilisateur.

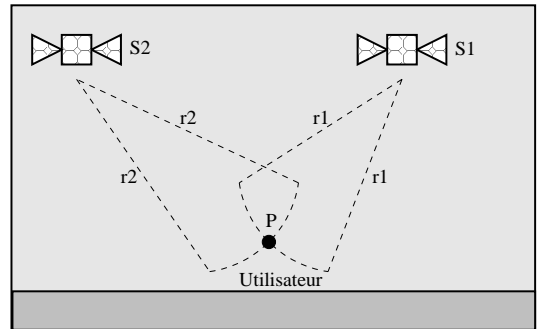


FIG. 5.4 – *Émission d'un signal*

5.3 Détail des calculs

Nous allons détailler le calcul de la position de l'utilisateur dans le cas d'une linéarisation du problème au voisinage de l'utilisateur ($M = (X, Y, Z, c\tau)$). On approche en effet la sphère par un plan tangent et on calcule l'intersection de ces plans.

En ce qui concerne le nombre d'équations et étant donné que les estimations sont entachées d'erreurs, on utilise un estimateur moindres carrés.

On a donc :

$$r'_i(M) = \sqrt{(X - X_i)^2 + (Y - Y_i)^2 + (Z - Z_i)^2} + c\tau \quad (5.7)$$

Soit M_0 un point proche de la solution :

$$r'_i(M) = r'_i(M_0) + \frac{\partial r'_i(M_0)}{\partial x}(X - X_0) + \frac{\partial r'_i(M_0)}{\partial y}(Y - Y_0) + \frac{\partial r'_i(M_0)}{\partial z}(Z - Z_0) + \frac{\partial r'_i(M_0)}{\partial(c\tau)}c(\tau - \tau_0) \quad (5.8)$$

$$\frac{\partial r'_i(M_0)}{\partial x} = \frac{(X - X_i)}{\sqrt{(X - X_i)^2 + (Y - Y_i)^2 + (Z - Z_i)^2}} \quad (5.9)$$

$$\frac{\partial r'_i(M_0)}{\partial y} = \frac{(Y - Y_i)}{\sqrt{(X - X_i)^2 + (Y - Y_i)^2 + (Z - Z_i)^2}} \quad (5.10)$$

$$\frac{\partial r'_i(M_0)}{\partial z} = \frac{(Z - Z_i)}{\sqrt{(X - X_i)^2 + (Y - Y_i)^2 + (Z - Z_i)^2}} \quad (5.11)$$

$$\frac{\partial r'_i(M_0)}{\partial(c\tau)} = 1 \quad (5.12)$$

On pose G_0 la matrice des dérivées partielles qui est une matrice $n \times 4$ (n étant le nombre de satellites en visibilité).

$$G_0 = \begin{bmatrix} \frac{\partial r'_1(M_0)}{\partial x} & \frac{\partial r'_1(M_0)}{\partial y} & \frac{\partial r'_1(M_0)}{\partial z} & \frac{\partial r'_1(M_0)}{\partial(c\tau)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r'_n(M_0)}{\partial x} & \frac{\partial r'_n(M_0)}{\partial y} & \frac{\partial r'_n(M_0)}{\partial z} & \frac{\partial r'_n(M_0)}{\partial(c\tau)} \end{bmatrix} = - \begin{bmatrix} e_1^{0T} & 1 \\ \vdots & \vdots \\ e_n^{0T} & 1 \end{bmatrix} \quad (5.13)$$

e_i étant le vecteur directeur unitaire de la droite utilisateur - satellite i .

$$r'(M) = r'(M_0) - G_0(M - M_0) \quad (5.14)$$

$$r'_i(M) = \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot \begin{bmatrix} r'_i(M_0) \end{bmatrix} - \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot M + \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} M_0 \quad (5.15)$$

$$r'_i(M) = \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot \begin{bmatrix} r'_i(M_0) \end{bmatrix} - \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot M + \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \begin{bmatrix} r_u(M_0) \\ c\tau_0 \end{bmatrix} \quad (5.16)$$

$$r'_i(M) = \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot \begin{bmatrix} r'_i(M_0) \end{bmatrix} - \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot M + \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \begin{bmatrix} r'_u(M_0) \end{bmatrix} \quad (5.17)$$

$r'_u = (X, Y, Z, c\tau)$ est le vecteur position de l'utilisateur avec biais d'horloge.

$$r'_i(M) = \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot \begin{bmatrix} r'_i(M_0) + r'_u(M_0) \end{bmatrix} - \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot M \quad (5.18)$$

$$r'_i(M) = \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot \begin{bmatrix} R'_i \end{bmatrix} - \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot M \quad (5.19)$$

La composante $R'_i(X_i, Y_i, Z_i, c\tau)$ est le vecteur position du i^{eme} satellite avec biais d'horloge.

$$r'_i(M) = D'_i - \begin{bmatrix} e_i^{0T} & 1 \end{bmatrix} \cdot M \quad (5.20)$$

La composante D'_i est la projection de R'_i sur la droite utilisateur - satellite i .

$$r'(M) = D' - G_0 \cdot M \quad (5.21)$$

En passant aux estimations :

$$\hat{r}(M) = \hat{D} - G_0 \cdot M \quad (5.22)$$

les composantes \hat{r}_i du vecteur \hat{r} sont les pseudo-distances estimées de l'utilisateur au i^{eme} satellite (avec erreur de mesure).

$$\epsilon = (\hat{D} - \hat{r}) = G_0 M \quad (5.23)$$

Les composantes ϵ_i du vecteur ϵ sont les erreurs de mesure qui suivent une loi $N(\mu_i, \sigma_i)$. On suppose ici que $\mu_i = 0$ et $\sigma_i = \sigma$ pour tous les satellites.

On estime M à l'aide des moindres carrés et on obtient :

$$\widehat{M} = (G_0^T G_0)^{-1} G_0^T \epsilon = G_0^* \epsilon \quad (5.24)$$

G_0^* étant la matrice pseudo-inverse de G .

On constate que la valeur de \widehat{M} est fortement conditionnée par celle de M_0 . Pour calculer avec une précision quasi indépendante de M_0 la valeur de \widehat{M} , on met en place un algorithme itératif ([54]) initialisé avec M_0 et tel qu'à l'itération k on calcule :

$$\widehat{M}_k = (G_{k-1}^T G_{k-1})^{-1} G_{k-1}^T \epsilon \quad (5.25)$$

avec

$$G_{k-1} = - \begin{bmatrix} e_1^{k-1T} & 1 \\ \vdots & \vdots \\ e_n^{k-1T} & 1 \end{bmatrix} \quad (5.26)$$

e_i^{k-1} étant le vecteur directeur unitaire de la droite \widehat{M}_{k-1} -Satellite i .

La convergence est déterminée par un test de stabilité (par exemple lorsque $\|\widehat{M}_k - \widehat{M}_{k-1}\|$ est inférieur à un seuil fixé (10^{-2})).

La position estimée de l'utilisateur est alors la dernière valeur \widehat{M}_k calculée.

La vitesse de convergence est fortement conditionnée par le choix du point de départ (C'est une des difficultés de l'algorithme). Plus le point de départ est proche de la position réelle de l'utilisateur plus l'algorithme convergera rapidement. On prend généralement comme point de départ M_0 , la dernière position calculée.

5.4 Évaluation du critère d'optimisation

5.4.1 Introduction

Le critère utilisé pour optimiser une constellation dédiée à la navigation est basé sur le calcul d'erreur relatif au positionnement d'un utilisateur sur la terre : on parle de **dilution de précision (Dop)**.

Pour réaliser cette évaluation, chaque point du maillage (chapitre 3) représente un utilisateur potentiel. On suppose donc connue la position de l'utilisateur dans l'algorithme de triangulation. La position des satellites et la loi sur l'erreur de mesure sont également connues.

On peut donc calculer la matrice G et la matrice M en une seule itération (L'algorithme n'est pas itératif puisque le point de départ est le point de convergence). Par contre le problème reste sur-déterminé et on utilise l'estimateur des moindres carrés pour connaître l'erreur commise lors du calcul de la position.

5.4.2 Les différents critères utilisables

Pour évaluer une constellation dédiée à la navigation, on utilise les critères suivants qui expriment la précision du positionnement en fonction des positions relatives de l'utilisateur et des satellites. Ce sont les critères les plus utilisés ([7],).

1. GDOP = $\frac{1}{\sigma} \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + \sigma_b^2}$ = Geometric Dilution Of Precision
2. PDOP = $\frac{1}{\sigma} \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$ = Position Dilution Of Precision
3. HDOP = $\frac{1}{\sigma} \sqrt{\sigma_x^2 + \sigma_y^2}$ = Horizontal Dilution Of Precision
4. VDOP = $\frac{\sigma_z}{\sigma}$ = Vertical Dilution Of Precision
5. TDOP = $\frac{\sigma_b}{\sigma}$ = Time Dilution Of Precision

$\sigma_x, \sigma_y, \sigma_z$ étant l'écart type des erreurs d'estimation sur la position de l'utilisateur et σ_b l'écart type biais sur l'horloge. L'unité utilisée est le mètre.

Dans l'expression des différents critères, on suppose que les erreurs de mesure sont indépendantes, de moyenne nulle et d'écart type σ . Dans le cas réel, il n'y a pas indépendance car l'écart type est fonction de l'angle d'élévation du satellite. On est amené à traiter une matrice de covariance pleine. De plus on n'a pas les mêmes écart-types pour tous les satellites.

5.4.3 Expression du critère en fonction des données de l'algorithme

Après avoir calculé la matrice \widehat{M} . On calcule la matrice de covariance de \widehat{M} , $cov(\widehat{M})$.

$$cov(\widehat{M}) = (G^T \cdot G)^{-1} \cdot G^T \cdot cov(\epsilon) \cdot G \cdot (G^T \cdot G)^{-T} \quad (5.27)$$

Si on considère que toutes les erreurs sont indépendantes, de moyenne nulle et d'écart type σ on obtient :

$$cov(\epsilon) = \sigma^2 \cdot I \quad (5.28)$$

avec I la matrice identité.

D'où :

$$cov(\widehat{M}) = \sigma^2 (G^T \cdot G)^{-1} \quad (5.29)$$

σ est la covariance des erreurs de mesure et $((G^T G)^{-1})$ indique la contribution de l'erreur de restitution de la géométrie des satellites.

La matrice $(G^T \cdot G)^{-1}$ étant symétrique définie positive, elle est diagonalisable en une matrice D telle que :

$$D = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 \\ 0 & 0 & 0 & \sigma_b^2 \end{bmatrix} \quad (5.30)$$

et $trace((G^T \cdot G)^{-1}) = trace(D)$.

Les différents critères ont donc l'expression suivante :

- $GDOP = \sqrt{trace((G^T G)^{-1})}$
- $PDOP = \sqrt{\sum_{i=1}^3 (G^T G)_{ii}^{-1}}$
- $HDOP = \sqrt{\sum_{i=1}^2 (G^T G)_{ii}^{-1}}$
- $VDOP = \sqrt{(G^T G)_{33}^{-1}}$
- $TDOP = \sqrt{(G^T G)_{44}^{-1}}$

5.4.4 Configuration optimale

Cas 2D

En 2D on a trois inconnues (x , y et b). Il faut donc au moins trois satellites pour résoudre le problème. On montre assez facilement que pour minimiser le GDOP (on souhaite avoir l'erreur la plus faible possible), il faut optimiser la géométrie des satellites par rapport à l'utilisateur. La position optimale est telle que les points A, B et C forment un triangle équilatéral autour de l'utilisateur. En effet, le GDOP est inversement proportionnel à la surface de ce triangle et le triangle équilatéral a la plus grande surface.

Dans le cas où l'on dispose de plus de trois satellites, l'équirépartition des satellites sur le cercle est la meilleure répartition.

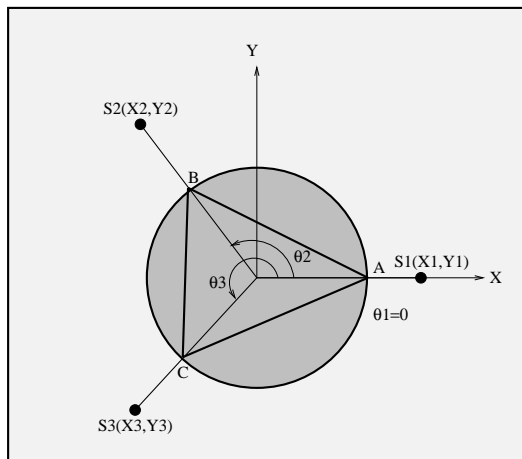


FIG. 5.5 – Calcul du GDOP en 2D

Cas 3D

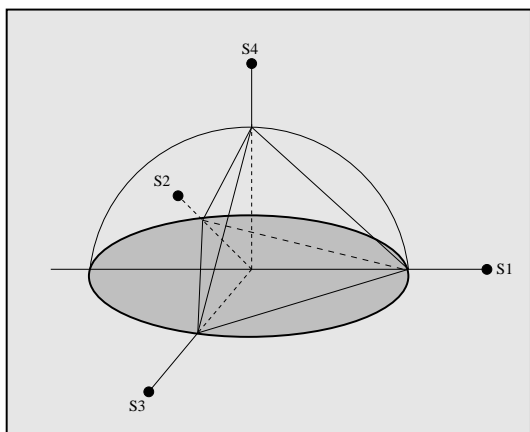


FIG. 5.6 – Calcul du GDOP en 3D

C'est le même principe que précédemment. Ayant quatre paramètres inconnus (x , y , z et b), il faut au moins quatre satellites en vue. La configuration optimale consiste à placer un satellite à la verticale de l'utilisateur et à placer les autres satellites dans un même plan orthogonal à la ligne utilisateur-satellite de façon à obtenir un tétraèdre régulier.

5.5 Intégration des pannes

Lorsque l'on dimensionne et configure une constellation, on doit souvent prendre en compte le cas où certains satellites tomberaient en panne. Étant donné le temps nécessaire pour remplacer un satellite (chapitre 3, page 51), il faut dimensionner la constellation de manière à ce qu'elle puisse être exploitable (même en mode dégradé), pendant le remplacement du satellite. La constellation doit donc être **robuste**. Le calcul de robustesse consiste à évaluer l'impact des pannes sur les performances du système et donnera lieu à un nouveau critère : **La disponibilité** [79].

La disponibilité d'une constellation Cs est évaluée localement en un point de maille m (notée $Dispo(Cs,m)$) et représente le pourcentage de temps pendant lequel les performances minimales attendues de Cs sont vérifiées en m ($PDOP < 6$ par exemple). Elle doit prendre en compte tous les cas de panne et sa valeur (en chaque maille) est une combinaison linéaire des performances calculées en m pour chaque cas de panne ([101]). Le poids affecté à chacune des performances dépend de la probabilité pour que la panne considérée se réalise. La performance minimale attendue est un calcul lié au domaine d'application (télécommunications, navigation) et rend compte du calcul d'un critère qui peut être un taux de couverture ou encore un calcul de $xDop$, x mis pour G,H,V , ou P . Une fois la disponibilité calculée en chaque point de maille, un post-traitement est nécessaire pour affecter une unique valeur à la constellation. On calcule généralement le pire cas de disponibilité (critère utilisé en navigation où les contraintes sont très fortes cf. [73]), ou la moyenne sur toute la simulation (on parle de **constellation value (CV)**, [79]).

Le calcul de la disponibilité de la constellation Cs en une maille m est donné par l'équation 5.31 :

$$Dispo(Cs,m) = \sum_{E \in \Pi(Cs)} P(E) \cdot \frac{1}{T} \sum_{t=1}^T f(Cs,m,t,E) \quad (5.31)$$

$f(Cs,m,t,E)$ est une fonction qui renvoie 1 si la performance minimale de Cs à l'instant t en m est assurée sachant que l'ensemble des satellites E est en panne. $P(E)$ est la probabilité pour que l'ensemble des satellites E soient en panne. Si on note p_i la probabilité pour que le satellite S_i soit en panne alors $P(E) = \prod_{S_i \in E} p_i \times \prod_{S_i \notin E} (1 - p_i)$.

On calcule la disponibilité de manière adroite en tenant compte de la visibilité des satellites à chaque instant et en chaque point de maille. Il faut enfin noter que la prise en compte des pannes ne fait qu'amplifier l'évaluation d'un critère déjà coûteux en temps de calculs.

Conclusion

L'évaluation des critères est un facteur limitant dans la recherche de constellations. Les temps de calculs prohibitifs associés à la simulation, à l'évaluation des performances locales et à la prise en compte de pannes rapportés à la taille de l'espace de recherche présentent une contrainte majeure que doivent prendre en compte les algorithmes de recherche.

Toute amélioration apportée aux techniques d'évaluation des constellations (simplification des critères, diminution des périodes de simulation, ...) se répercutera immédiatement sur les performances de l'algorithme, quel qu'il soit.

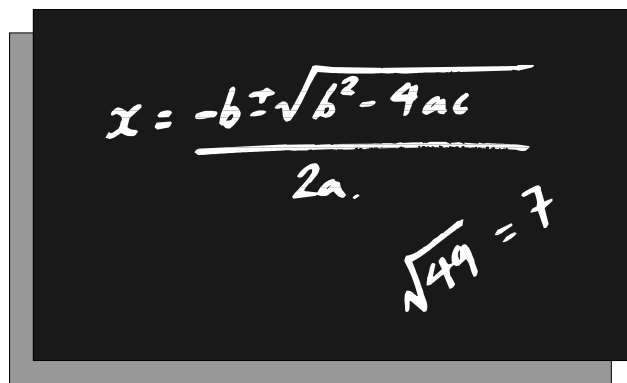
Nous verrons dans le cadre de la modélisation du problème et dans la construction de l'algorithme proposé quels sont les efforts qui ont été mis en oeuvre dans ce sens.

De plus, l'impossibilité d'exprimer les critères de manière analytique proscrit l'utilisation d'algorithmes classiques pour la résolution du problème global et nous conduit tout naturellement vers des techniques de recherche basées sur des heuristiques. Ces méthodes que nous utilisons ont d'ailleurs déjà été exploitées dans le cadre de problèmes issus du monde spatial ([99], [98]).

Par ailleurs, on constate un certain nombre de manques dans les systèmes actuels. Si on considère par exemple la constellation GPS, qui fournit un service de localisation d'un utilisateur sur la terre, et les contraintes de la navigation civile en terme de précision de la position en phase de décollage ou d'atterrissage, celle-ci ne permet pas de satisfaire toutes les contraintes. Des systèmes au sol ([10]) sont donc mis en place pour augmenter la précision autour des aéroports. Il apparaît donc utile de définir un autre système de navigation par satellite qui permettrait de satisfaire les précisions requises par la navigation aérienne et ce, sans système annexe au sol.

Partie III

Les méthodes d'optimisation utiles dans la suite de l'étude


$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
$$\sqrt{49} = 7$$

Nous avons vu au cours des deux précédentes parties que des méthodes d'optimisation classiques ne pouvaient être mises en oeuvre pour la résolution du problème global (non régularité du critère, ...).

Nous sommes donc naturellement guidés vers des méthodes sous-optimales pour résoudre le problème.

Par ailleurs, on note dans la littérature un couplage fort entre les modélisations et les résolutions proposées ce qui nous conduit à présenter d'abord de manière théorique les méthodes d'optimisation.

Nous présenterons dans un premier temps le cadre théorique de l'optimisation avant de nous intéresser aux algorithmes génétiques (chapitre 7) qui seront utilisés lors de la présentation de l'état de l'art en matière de design de constellations. Nous présenterons ensuite la recherche tabou (chapitre 8) et l'algorithme de recherche de plus grande pente (chapitre 9) qui seront utilisés dans notre travail.

Chapitre 6

Introduction

La recherche d'une solution optimale à un problème donné peut prendre une multitude de formes en fonction des hypothèses que l'on pose et de la modélisation que l'on a réalisé. Prenons le problème (P) sous sa forme la plus générale (6.1). On cherche la valeur de x , prenant ses valeurs dans l'espace E et soumis aux contraintes C , qui minimise la fonction f .

$$(P) \begin{cases} \text{Min}(f(x)) \\ x \in C \subset E \end{cases} \quad (6.1)$$

Nous allons commencer par définir un certain nombre de termes que nous utiliserons dans la suite du chapitre.

Définition 1 (Solution globale) \bar{x} solution globale de $(P) \iff \forall x \in C f(\bar{x}) \leq f(x)$

Définition 2 (Solution locale) \bar{x} solution locale de $(P) \iff \exists V$ voisinage de \bar{x} telque $\forall x \in V \cap C f(\bar{x}) \leq f(x)$

Les techniques d'optimisation classiques requièrent souvent des hypothèses strictes sur l'espace de recherche (E) et la fonction d'évaluation des solutions (f). Ces méthodes, basées sur une étude analytique de la fonction d'évaluation, ont besoin d'hypothèses de régularité (continuité, dérivabilité) sur f . On trouve dans ce cas les deux conditions nécessaires suivantes qui permettent de chercher une solution au problème.

Condition nécessaire du premier ordre

Si f est dérivable sur un ouvert de E et admet un minimum local en \bar{x} alors : $f'(\bar{x}) = 0$.

Condition nécessaire du second ordre

Si f est dérivable sur un ouvert de E et admet un minimum local en \bar{x} alors : $f'(\bar{x}) = 0$ et $f''(\bar{x})(\phi, \phi) \geq 0, \forall \phi \in E$.

En outre on peut parfois accéder à des conditions suffisantes. **Conditions suffisantes d'existence de solution**

Si f est dérivable sur un ouvert de E et deux fois dérivable en \bar{x} et si $f'(\bar{x}) = 0$ alors : si $\exists \alpha > 0 | f''(\bar{x})(\phi, \phi) \geq \alpha \|\phi\|^2, \forall \phi \in E$, f admet un minimum local strict en \bar{x} .

Si f est deux fois dérivable sur un ouvert de E et si $f'(\bar{x}) = 0$ alors si il existe une boule B centrée en \bar{x} telle que : $f''(x)(\phi, \phi) > 0, \forall \phi \in E, \forall x \in B$, f admet un minimum local strict en \bar{x} .

Sous de telles hypothèses, des méthodes comme la méthode de Newton, la méthode des régions de confiance, ou encore la méthode des gradients conjugués donnent de bons résultats ([49]). En effet, des propriétés sur l'existence de solutions et sur la convergence des méthodes assurent de trouver un optimum (local ou global) au problème. Lorsque l'on est assuré de converger vers un optimum global de la fonction d'évaluation, on parle de **méthode d'optimisation globale** dans le cas contraire on parle de **méthode d'optimisation locale**.

Mais qu'en est-il lorsque les hypothèses précédentes sur l'espace de recherche et/ou sur la fonction d'évaluation ne sont plus vérifiées? Pour traiter de tels problèmes les méthodes classiques ne sont plus adaptées et il faut faire appel à des techniques de recherche qui font abstraction d'exigences sur l'espace de recherche et sur la fonction d'évaluation. Ces méthodes, plus ou moins complexes, ont à faire face aux mêmes problèmes de convergence locale ou globale que les méthodes classiques. Ne pouvant se baser, dans la définition des grands principes de ces méthodes, sur des propriétés de l'espace de recherche ou de la fonction d'évaluation, des techniques d'exploration de diverses régions de l'espace E sont mises en place pour assurer la recherche d'un optimum global : on parle de **diversification** de la recherche. D'un autre côté, la convergence (souvent asymptotique) vers l'optimum est assurée par d'autres techniques : on parle d'**intensification** de la recherche.

Dans le contexte qui nous intéresse, celui du design de constellation de satellite, nous avons à prendre en compte un certain nombre de paramètres tels que le **temps de calcul** et la **taille de l'espace de recherche**. Des méthodes simplistes, telles que la **recherche exhaustive** ou encore la **recherche aléatoire**, applicables lorsque l'espace de recherche n'est pas très vaste et la fonction d'évaluation peu coûteuse en temps de calcul, ne peuvent être appliquées à notre problème.

Deux grandes familles de méthodes peuvent potentiellement satisfaire notre problème. Toutes deux mettent en oeuvre le processus d'intensification et de diversification mais de manière très différente. Il s'agit dans ce cadre de faire évoluer une population de solutions. D'un côté on trouve les **méthodes stochastiques** qui commencent par une forte diversification en mettant en jeu un ensemble de solutions réparties sur l'espace de recherche avant de concentrer leurs efforts vers des régions prometteuses (intensification) : on parle de **méthode d'inspiration globale**. Nous étudierons dans ce cadre les **algorithmes génétiques (AG)** très représentatifs de ces méthodes et des approches classiques en matière d'optimisation de constellations de satellites. D'un autre côté on trouve les **méthodes heuristiques** de recherche par voisinage qui permettent de réduire l'espace de recherche en tenant compte d'un certain nombre de propriétés des solutions et des voisinages de solutions. Ces méthodes sont elles **d'inspiration locale**, c'est à dire quelles mettent en jeu une unique solution que

l'on fait évoluer. Les phases d'intensification et de diversification sont alternées afin que la solution explore (intensification) plusieurs régions (diversification) de l'espace de recherche. Nous étudierons dans ce cadre la **recherche tabou** dont les aspects avancés intègrent des techniques performantes de convergence (oscillations stratégiques, ...) et que nous utiliserons pour résoudre le problème de design de constellations de satellites. D'autres algorithmes de recherche heuristique par voisinage, proches de la recherche tabou, (recuit simulé ([70],[71]), VNS ([62]), ...) pourraient être également envisagés. Néanmoins le cadre de la recherche tabou nous semble assez général pour correctement mesurer la pertinence de ces approches dans le contexte de l'optimisation de constellations.

Il faut noter que ces deux méthodes ont, au mieux, une assurance statistique de convergence : la probabilité de tendre vers la solution globale du problème tend au mieux (sous de bonnes hypothèses) vers 1 lorsque le nombre d'itérations tend vers l'infini ([17], [45]). Il n'y a donc plus de résolution exacte du problème.

La base des méthodes que nous allons présenter est un algorithme "idiot" et peu performant : la recherche aléatoire.

```

 $x^c$  = solution choisie aléatoirement dans  $C$ 
 $E$  tantque on n'est pas satisfait
     $x^+$  = solution choisie aléatoirement dans  $C$ 
    si  $f(x^+) < f(x^c)$  alors
         $x^c \leftarrow x^+$ 
    fin tantque

retourner  $x^c$ 

```

TABLE 6.1 – *Algorithme de recherche aléatoire*

L'intérêt d'un tel algorithme est qu'il fonctionne pour tous les critères et pour tous les espaces. L'obtention de l'optimum n'est assurée que par la probabilité (faible) de tirer aléatoirement l'élément. Les méthodes heuristiques sont des raffinements de cet algorithme qui consiste à choisir (ou calculer) la (les) nouvelle(s) solution(s) à partir de la (des) solution(s) courante(s) afin d'améliorer les résultats en termes de robustesse et de déterminisme par la mise en oeuvre d'heuristiques. Nous allons successivement présenter les algorithmes génétiques et la recherche tabou dans le cadre des méthodes heuristiques et la méthode de la plus forte pente dans le cadre des méthodes classiques. Cette dernière méthode sera utilisée pour résoudre une sous partie de notre problème avec un certain nombre d'hypothèses que nous définirons.

Chapitre 7

Les algorithmes génétiques

7.1 Introduction

Les algorithmes génétiques (AG) que nous allons présenter ici, font partie des méthodes stochastiques d'inspiration globale. Ces méthodes sont souvent utilisées lorsque l'on ne dispose pas d'informations sur les propriétés de la surface du critère (abstraction des propriétés sur la fonction d'évaluation).

Nous y ferons référence lorsque nous présenterons certaines méthodes utilisées pour le design de constellations de satellites.

Les AG sont issus, comme leur nom le laisse supposer, des techniques d'évolution naturelles. Les AG ont été appliqués en optimisation mathématique pour la première fois par John Holland [65] mais des problèmes matériels apparurent à cette époque en raison du coût élevé en temps de calcul de ces méthodes.

L'ouvrage de Goldberg [48] fait aujourd'hui office de référence car il synthétise et illustre les grands principes des AG. Les AG sont des algorithmes stochastiques d'optimisation globale. Le principe consiste à faire évoluer un ensemble de solutions avec des règles proches de celles qui régissent l'évolution naturelle. On explore donc simultanément plusieurs parties de l'espace de recherche. Cette répartition des solutions assure la diversification de la recherche, phase nécessaire à l'obtention d'un optimum global. La phase d'intensification est assurée dans un second temps par des opérateurs que nous détaillerons ultérieurement pour affiner la recherche dans les régions les plus prometteuses ([95]).

7.2 Notations et principe

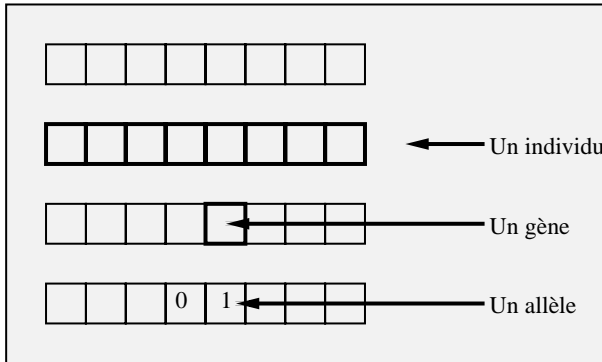


FIG. 7.1 – *Inspiration génétique*

Nous ferons cette hypothèse sur le codage d'un individu dans le reste de ce chapitre.

Nous allons présenter au cours de ce chapitre les règles qui régissent l'évolution de cette population. Tout d'abord, la population initiale est générée aléatoirement afin de couvrir l'espace de recherche de manière uniforme. Elle constitue la génération G_0 .

On passe de la génération G_i à la génération G_{i+1} en appliquant trois opérateurs sur la génération courante G_i : la sélection, le croisement et la mutation.

Ces trois opérateurs sont complémentaires et assurent à la fois un brassage des individus et une exploitation des meilleurs d'entre eux.

L'opérateur de **sélection** permet de favoriser les meilleurs éléments de la génération courante qui serviront de géniteurs.

L'opérateur de **croisement** permet un brassage entre les gènes des parents afin de créer des individus nouveaux formant la nouvelle génération.

Enfin, l'opérateur de **mutation** assure une certaine diversification en changeant aléatoirement la valeur d'un gène d'un individu de la nouvelle génération.

Dans le reste de cette section on considère le problème suivant : soit $P_g = \{x_g^i\}_{i \in [1, n]}$ la population de la génération g composée de n individus. Le problème d'optimisation tend à chercher l'individu \hat{x} qui maximise la fonction F d'évaluation.

La représentation des solutions est calquée sur la génétique.

On parle de **population d'individus**, un individu est représenté par un **chromosome**, et chaque chromosome est composé de **gènes**, ayant comme ensemble de valeurs potentielles des **allèles**.

Chaque individu est évalué en utilisant une fonction critère que l'on cherche à maximiser. La figure 7.1 illustre ces notations à l'aide d'un chromosome représenté par un vecteur de bits ([103]).

7.3 Les règles d'évolution

7.3.1 La sélection

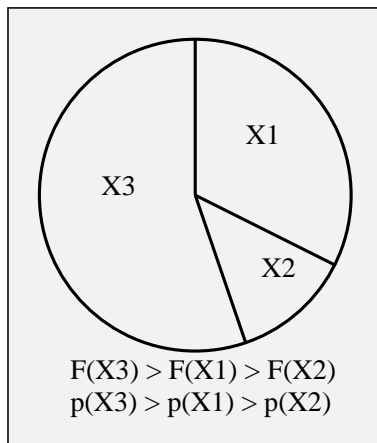


FIG. 7.2 – Mécanisme de sélection par roulette

Ce mécanisme de sélection assure l'exploitation des bons individus et donc la convergence des générations vers une meilleure population.

7.3.2 Le croisement

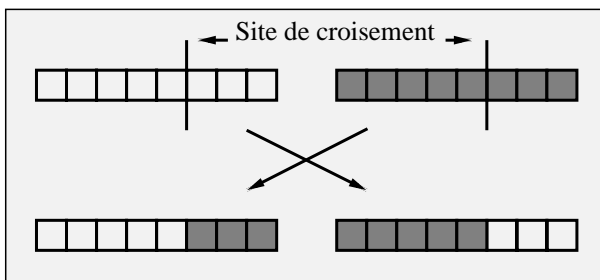


FIG. 7.3 – Mécanisme de croisement

aléatoirement (ou avec des probabilités dépendant du problème) un site de croisement. On opère ensuite une permutation entre deux mêmes parties des parents pour générer les chromosomes fils x_{g+1}^{ij1} et x_{g+1}^{ij2} .

Comme nous l'avons vu, les algorithmes génétiques sont basés sur les règles d'évolution naturelle. On a donc besoin, pour produire la génération suivante, de sélectionner des individus qui serviront de géniteurs. On réalise ici une sélection probabiliste par un tirage aléatoire non uniforme sur l'ensemble de la population. C'est-à-dire que plus l'évaluation d'un individu est élevée, plus on a de chances de le sélectionner. Ce mécanisme est qualifié de **tirage par roulette** [48].

La probabilité de sélection de l'individu x_g^i est généralement donnée par la probabilité $p_g^i = \frac{F(x_g^i)}{F_g}$ avec $F_g = \sum_{i=1}^n F(x_g^i)$.

Après la sélection des parents, il faut construire la génération suivante. Pour présenter un intérêt et faire progresser la recherche, la nouvelle génération doit comporter des individus différents de la précédente. Pour ce faire, l'opérateur de croisement assure une recombinaison des parents permettant un brassage de leurs informations génétiques ([97]).

On réalise donc un **croisement** entre deux des individus x_g^i et x_g^j sélectionnés et on détermine

Ce mécanisme peut être généralisé au cas de plusieurs parents, les croisements sont dans ce cas plus compliqués et on réalise des combinaisons pondérées entre les différents géniteurs. De même, il peut exister plusieurs sites de croisement permettant d'étendre le brassage, mais il faut toujours garder des proportions raisonnables pour ne pas tomber dans la génération d'individus aléatoires.

La méthode de croisement à utiliser dépend du problème à traiter mais il faut garder à l'esprit que le principe du croisement est un compromis et qu'il doit faire émerger des individus appartenant à des régions de l'espace de recherche différents de celles des parents pour assurer le brassage (principe de diversification). Mais dans le cas de deux géniteurs proches structurellement, il faut garantir que l'opérateur fournisse des descendants proches des parents.

7.3.3 La mutation

L'autre opérateur caractérisant les AG est l'opérateur de mutation. Il opère en réalisant une modification aléatoire de la valeur d'un gène. Chaque gène est soumis après croisement à un tirage aléatoire de probabilité p_m (probabilité de mutation) très faible. Dans le cas d'un tirage positif, le gène est muté par une règle arbitraire qui peut être :

- un tirage aléatoire de la valeur,
- l'ajout d'une perturbation à la valeur courante,
- la duplication d'un autre gène,
- ...

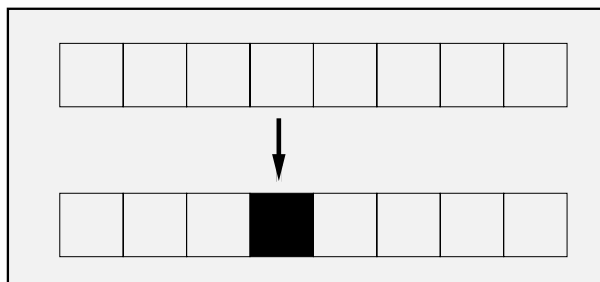


FIG. 7.4 – Mécanisme de mutation

Cet opérateur permet d'assurer une certaine diversification dans le sens où on peut potentiellement atteindre n'importe quel individu de l'espace de recherche à partir de n'importe quel autre (principe de diversification). Mais la probabilité de mutation étant faible on peut également engendrer des individus très proches de l'individu source et ainsi réaliser une exploration locale (principe d'intensification).

Là encore, l'opérateur est à appliquer en fonction du problème traité pour ne pas être confronté à des aberrations structurelles.

7.4 La mise en oeuvre des algorithmes génétiques

7.4.1 Le codage des individus et les paramètres de l'algorithme

Une des difficultés de la mise en oeuvre des AG est son paramétrage. En effet, la taille de la population peut avoir une grande influence sur la résolution du problème, tant au niveau temps de calcul qu'au niveau pertinence de la solution trouvée.

De même les probabilités de sélection, de croisement et de mutation peuvent changer la manière de parcourir l'espace de recherche.

Enfin, les individus peuvent subir différents codages avant d'être présentés à l'algorithme. Le codage est introduit pour faciliter les recombinaisons lors de la phase de croisement.

Il est fondamental de bien considérer cette étape pour ne pas avoir des solutions structurellement incorrectes après le croisement.

7.4.2 La convergence

La convergence des AG est assurée par les opérateurs de sélection et de croisement. Les premières générations sont suffisamment hétérogènes pour que le croisement assure la diversification, mais au fur et à mesure des générations, la population tend à s'uniformiser dans une région de l'espace de recherche que les croisements ne bouleversent plus.

7.4.3 La condition d'arrêt

Le condition d'arrêt de l'AG peut être de deux natures :

- soit on fixe un nombre maximal de générations à atteindre avant d'arrêter. Ceci assure une résolution en un temps borné.
- soit on attend la stabilité de la génération. C'est à dire lorsque les variations sont faibles en termes de propriétés des individus et/ou en termes de valeurs des solutions.

7.4.4 L'algorithme général

<p>Génération d'une population initiale P_0. <u>tantque</u> condition d'arrêt non satisfaite Évaluer le critère de chaque individu. Sélectionner les individus parents. Appliquer l'opérateur de croisement. Appliquer l'opérateur de mutation. Construire la nouvelle génération. <u>fin tantque</u></p>

TAB. 7.1 – *Algorithme génétique*

7.5 Aspects avancés des algorithmes génétiques

7.5.1 Fonction de pénalisation

Les **paramètres** (gènes) dont on doit fixer la valeur (prise dans l'alphabet d'allèles) sont souvent **contraints**. Les contraintes peuvent être d'**ordre 1** (appliquées directement à un seul paramètre) ou d'**ordre supérieur** (liant plusieurs paramètres).

Si les contraintes permettent souvent de réduire l'espace de recherche, elles augmentent par ailleurs la complexité des calculs. Pour tenir compte de ces contraintes on peut adopter plusieurs points de vue en fonction de la complexité de leur mise en oeuvre :

1. Si les **contraintes** sont **simples** (intervalle borné par exemple), on peut réaliser des opérateurs de croisement et de mutation qui donnent directement des solutions réalisables.
2. Si les **contraintes** sont **compliquées** (arité supérieure à 2), un certain nombre de calculs sont nécessaires pour valider une solution. On peut ici adopter deux méthodes [93],[61],[89]:
 - La première respecte les contraintes et n'autorise aucun débordement : si une contrainte est violée, la solution est soit rejetée, soit modifiée pour respecter les contraintes (utilisation d'une **fonction réparatrice**). L'avantage de cette méthode est que l'on reste constamment dans le domaine des solutions réalisables. Mais il faut évaluer le coût en calculs sur un nouvel individu de remplacement. En effet, rejeter la solution entraîne souvent un nouveau calcul du critère ce qui peut s'avérer honéroux. De même la réparation

doit se faire avec un minimum de calculs (on doit obtenir une solution "voisine" mais réalisable).

- La deuxième solution autorise l'algorithme à explorer temporairement des solutions violant certaines contraintes afin d'éviter de trop lourds calculs et permettre, sous certaines hypothèses, de converger plus vite.

Le non-respect d'une contrainte entraîne une **pénalité** fonction de l'amplitude du viol : on parle de **fonction de pénalité**. Cette fonction agit directement sur la *fonction d'évaluation* afin d'en modifier la valeur pour traduire ce non respect des contraintes. Typiquement, on diminue la valeur de la *fonction d'évaluation* pour que l'individu soit considéré comme moins bon. L'avantage de cette méthode est que l'on garde toutes les solutions trouvées, ce qui allège les calculs. Mais on peut ne pas converger vers une solution réalisable si on ne prend pas garde à l'évolution "à moyen terme" de ces solutions. Il faut en effet contraindre ces solutions à subsister un nombre fini d'itérations (si la solution n'a pas évoluée vers une solution réalisable, elle est supprimée).

Même si le domaine d'application des algorithmes génétiques est souvent l'optimisation non contrainte, on peut ainsi l'étendre à des problèmes fortement contraints. Les fonctions de pénalisation sont introduites lorsqu'il est difficile d'assurer qu'un croisement entre deux individus réalisables donnera un individu lui-même réalisable ou lorsqu'il est difficile de trouver d'emblée une solution réalisable.

7.5.2 La théorie des schémas

Les schémas sont des masques appliqués aux chromosomes pour les classer, afin de faire apparaître des similarités entre certaines solutions. Un schéma est représenté par un chromosome dont certains gènes ont une valeur fixée et d'autres peuvent prendre toutes les valeurs dans l'alphabet d'allèles. Le choix des schémas à adopter dépend du problème traité. La détermination des schémas intéressants dépend souvent de la représentation physique des gènes. On peut prendre des positions fixes pour des gènes que l'on sait déterminants dans l'évaluation d'un individu pour découvrir une bonne structure pour la solution. Un schéma s est caractérisé par deux paramètres :

- **L'ordre du schéma** $o(s)$: c'est le nombre de positions fixées.
- **La longueur du schéma** $\delta(s)$: c'est la distance entre la première position fixée et la dernière.

Ces schémas permettent également d'analyser les effets des opérateurs sur la population.

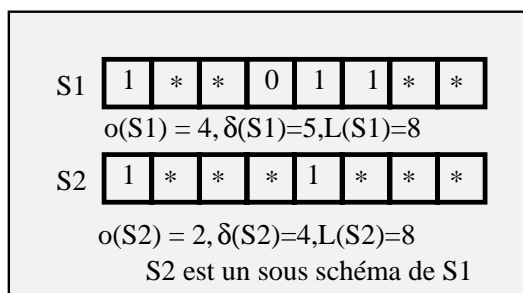


FIG. 7.5 - Exemple de schémas

Effet de la sélection

On suppose que le nombre de représentants du schéma s à la génération g est donné par $m(s,g)$, soit F_g^s , la valeur moyenne de la fonction d'évaluation sur l'ensemble des représentants du schéma s et \bar{F}_g la moyenne de f sur toute la population à la génération g . Alors on s'attend à ce qu'à l'itération $(g + 1)$ on ait :

$$m(s, g + 1) = m(s, g) \times \frac{F_g^s}{\bar{F}_g} \quad (7.1)$$

D'après cette relation, on s'attend à ce que certains schémas prennent une place de plus en plus importante (ceux dont l'évaluation moyenne est la meilleure) alors que d'autres vont disparaître. On voit donc ici que la connaissance du problème donne des informations sur l'évolution des solutions à travers les schémas. On peut même prendre le procédé à l'inverse, c'est à dire faire évoluer la population et analyser les regroupements qui s'effectuent pour en déduire des schémas et donc des propriétés sur les solutions.

Effet des croisements

Les croisements affectent les schémas puisque les sites de croisement peuvent se situer entre deux positions fixées et donc détruire le schéma d'un individu (ou au contraire en créer un nouveau). Si on introduit les croisements dans la relation 7.1, on obtient la relation suivante :

$$m(s, g + 1) \geq m(s, g) \cdot \frac{F_g^s}{\bar{F}_g} \left[1 - p_c \cdot \frac{\delta(s)}{l - 1} \right] \quad (7.2)$$

. avec p_c la probabilité de croisement et l la longueur du chromosome.

Effet de la mutation

Après application de l'opérateur de mutation la relation 7.2 devient :

$$m(s, g + 1) \geq m(s, g) \cdot \frac{F_g^s}{\bar{F}_g} \left[1 - p_c \cdot \frac{\delta(s)}{l - 1} - o(s)p_m \right] \quad (7.3)$$

avec p_m la probabilité de mutation.

Ces propriétés 7.1, 7.2 et 7.3 permettent de prouver le théorème des schémas : "le nombre de schémas courts, d'ordre faible, et mieux adaptés augmente selon une loi puissance au cours du temps".

7.5.3 Le scaling

Dans la population initiale, l'existence d'individus nettement supérieurs aux autres (mais pas forcément optimums) conduit généralement très vite à restreindre l'espace de recherche au voisinage de ces individus. Ainsi, lors de la sélection, et dans un souci de diversification, on souhaite

harmoniser les évaluations des individus. Pour ce faire, on utilise le *scaling* qui consiste à agir sur la fonction d'évaluation afin d'en modifier la valeur dans le sens souhaité. En effet, si pour les premières itérations on veut réduire les écarts d'évaluation, il est aussi très intéressant de pouvoir exagérer cet écart en fin d'algorithme pour accélérer la convergence.

Soit f' la fonction d'évaluation après le scaling et f celle avant le scaling. On distingue donc plusieurs types de scaling :

1. **Le scaling linéaire** : On a la relation $f' = af + b$. Généralement $a < 1$ ce qui réduit les écarts et favorise les explorations. L'inconvénient de ce scaling est qu'il est statique, ce qui pénalise la fin de la convergence lorsque l'on souhaite accélérer le processus.
2. **Le scaling exponentiel** : il permet de résoudre le problème soulevé précédemment en faisant intervenir dans le scaling l'indice g de la génération courante. L'expression de ce scaling est la suivante : $f = (f)^{k(g)}$. La fonction k permet d'adapter le coefficient de réduction-amplification en fonction de l'avancement de l'algorithme.
 - Pour $k(n) < 1$ on réalise une réduction des écarts d'évaluation (on favorise l'exploration).
 - Pour $k(n)$ voisin de 1, le scaling est inopérant.
 - Pour $k(n) > 1$, on réalise une amplification des écarts ce qui accélère la convergence.

Les phases **d'exploration et d'intensification** sont donc alternées au cours de la recherche mais pas de façon uniforme. On préférera une exploration plus importante en début d'algorithme pour ne pas se focaliser sur une solution locale, puis, lorsqu'une région intéressante est localisée, on se concentre sur la zone en réalisant une intensification afin de converger plus vite. De la même façon, lorsque des améliorations locales ne sont plus possibles, on réalisera une exploration d'autres régions.

7.5.4 Notion d'élitisme-tournoi

Lors de la génération des descendants, on n'est pas sûr d'obtenir de bons individus ainsi on peut souhaiter garder les meilleurs individus de la population courante afin de converger vers une population contenant des individus dominants : **c'est l'élitisme**.

Une version plus raffinée de l'élitisme consiste à réaliser des tournois entre individus de la nouvelle génération et individus de la génération courante. Un tournoi consiste à confronter la population de géniteurs à la population des descendants et à garder les meilleurs.

On réalise souvent un tournoi entre les enfants et les parents afin de ne pas générer (en probabilité) d'enfants moins bons que leurs parents. Mais si on ne garde que les deux meilleurs parmi les quatre concurrents, on risque de tomber dans le piège de maxima locaux. Pour contrer ce problème, on

réalise très souvent un tournoi de la manière suivante :

Sélectionner les deux parents
Générer les deux enfants.
Si le premier enfant est meilleur que les deux parents alors
 On le sélectionne pour la nouvelle génération
Sinon
 On le sélectionne avec une certaine probabilité
 dépendant du classement relatif de l'enfant par rapport aux deux parents.
Fin Si
Faire de même avec le second enfant.

TAB. 7.2 – *le tournoi*

7.5.5 Le sharing

Dans certains cas, lorsque la fonction d'évaluation présente un grand nombre de maxima locaux, il est intéressant de répartir une sous population sur chacun des sommets de la fonction d'évaluation. Cela permet de trouver, comme résultat de l'optimisation, une famille de solutions locales possédant des propriétés différentes. Le nombre d'individus sur un sommet est proportionnel à la fonction d'évaluation en celui-ci. Ce procédé, le **Sharing** ([4]), présente l'avantage de pouvoir paralléliser les traitements puisqu'ils tendent vers l'indépendance. On peut même lancer plusieurs algorithmes génétiques (un par sommet). Notons toutefois que des techniques sophistiquées de parallélisation existent.

La difficulté est ici de définir une distance permettant de séparer les différentes agrégation d'individus dans une région donnée. On introduit la notion de **cluster** centré en chaque région où des regroupements doivent être effectués (cf. [31]).

Après la définition de la distance, le problème est un problème de classification des individus dans les différents clusters.

7.6 Autres méthodes

Parmi les algorithmes évolutifs on distingue deux autres méthodes : la programmation évolutive ([8]) et les stratégies évolutives ([35]). Ces deux méthodes implémentent le concept d'algorithme

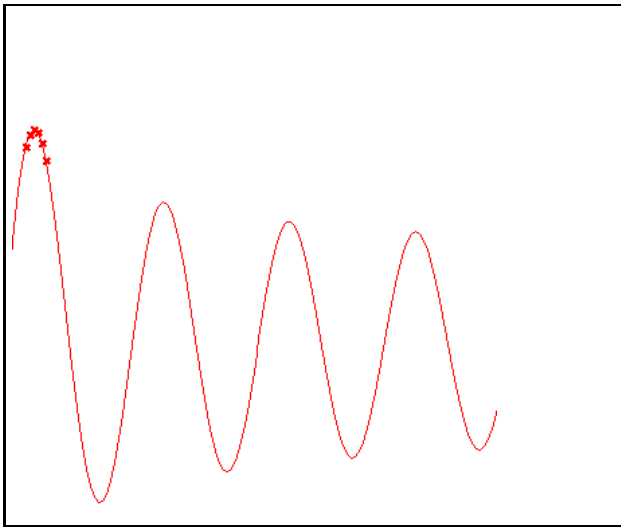


FIG. 7.6 – *Sans sharing*

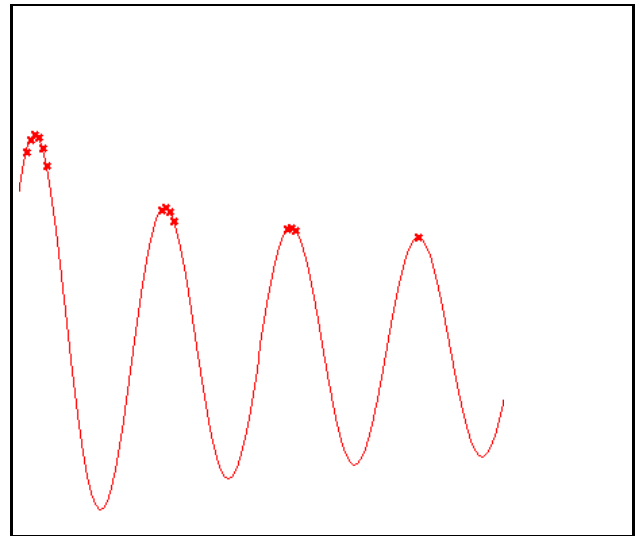


FIG. 7.7 – *Avec sharing*

évolutif de manière différente. Par exemple, dans la programmation évolutive, on ne met en oeuvre que la transformation des individus pour générer la nouvelle population (les perturbations dues à cette transformation sont suffisantes pour jouer le rôle de la recombinaison).

De même les méthodes dites de Scatter Search ([42], [44]) sont de même nature que ces approches évolutionnistes.

Chapitre 8

La recherche Tabou

8.1 Introduction

La recherche tabou (TS) est une méthode d'inspiration locale, c'est-à-dire qu'elle est basée sur l'évolution d'une unique solution dont on cherche un successeur dans le voisinage direct. La base des méthodes d'inspiration locale est un raffinement de la recherche aléatoire appelée **recherche par voisinage**.

Soit x^c la solution courante et $V(x^c)$ le voisinage de cette solution, on a l'algorithme suivant.

```
 $x^c$  = solution choisie dans  $C$   
tant que on n'est pas satisfait  
     $x^+$  = solution choisie aléatoirement dans  $V(x^c)$   
    si  $f(x^+) < f(x^c)$  alors  
         $x^c \leftarrow x^+$   
  
fin  
retourner  $x^c$ 
```

TAB. 8.1 – *Algorithme de recherche par voisinage*

Comparée à la méthode de recherche aléatoire, cette méthode a pour avantage de permettre un contrôle de la séquence des solutions. Le recuit simulé ([70], [71]) comme la recherche tabou ([40], [41]) sont fondés sur ce principe.

Ces méthodes sont assez récentes, par rapport aux méthodes classiques d'optimisation, puisque les articles de synthèse ([40], [41]) décrivant la recherche tabou date par exemple de 1989. Le principal

fondateur de cette méthode est *F. Glover*.

La recherche tabou, qui est un algorithme introduit dans le cadre de l'optimisation combinatoire, introduit des heuristiques permettant de choisir (ou de ne pas choisir) une solution voisine comme successeur de la solution courante.

On définit, pour accéder dans le voisinage V au successeur de la solution courante, un ensemble de **transitions** de manière heuristique (en fonction du problème traité). L'évolution de la solution correspond donc à un changement d'état de celle-ci en utilisant les différentes transitions. La méthode doit son nom à l'existence de transitions dites **Tabous**, c'est-à-dire que l'on ne peut pas les utiliser (à une itération donnée de l'algorithme) pour accéder au successeur. Cette notion d'interdiction (de Tabou) est très importante car elle permet de limiter le voisinage de la solution courante, et donc l'exploration, pour accélérer et guider stratégiquement la recherche. L'état tabou d'une solution ou d'une transition est fixé dynamiquement pendant la recherche en utilisant l'histoire de la recherche (solutions visitées, transitions empruntées, ...): on parle de **mémoire de l'algorithme** qui est un des autres aspects importants de la méthode.

Au cours de ce chapitre, nous allons détailler les principales caractéristiques de la recherche tabou en présentant les notions d'état tabou et de mémoire. Puis nous détaillerons quelques idées avancées telles que les oscillations stratégiques qui permettent de guider la recherche à un niveau supérieur.

8.2 Définitions

8.2.1 Transition avec attributs

On appelle **transition** le passage d'une solution x à une solution y et on la note $x \rightarrow y$. Pris comme tel, il existe autant de transitions que de paires ordonnées (x,y) dans l'espace de recherche E . Pour limiter les transitions applicables on définit la notion d'**attribut d'une transition**. Un attribut permet de caractériser un changement entre une solution x et une solution y , il caractérise donc la transition.

Prenons par exemple le cas de deux vecteurs binaires de dimension n . L'espace de recherche est donc $E = \{0,1\}^n$ et une solution a la forme suivante: $x = (x_1, \dots, x_n)$ avec $x_i \in \{0,1\}$. On note f la fonction d'évaluation.

Voici quelques exemples d'attributs que l'on peut définir pour une transition $x \rightarrow y$:

- $A_1 : x_i \neq y_i$.
- $A_2 : x_i = 0, y_i = 1$
- $A_3 : x_i = 1, y_i = 0$
- $A_4 : f(y) - f(x)$

– etc.

Pour chacun des attributs définis précédemment on définit un type de transition différent (Tr_1, Tr_2, Tr_3, Tr_4) et on ne qualifie de transition que les changements correspondant à l'un de ces types. On constate par ailleurs que les attributs peuvent être de nature qualitative (telle variable est modifiée) ou quantitative (l'incrément subi par la variable est de tant). Ces attributs servent de base pour l'application des métaheuristiques.

Habituellement, lorsque la taille du voisinage et le temps d'évaluation d'une solution le permettent, on choisit comme successeur de la solution courante la meilleure solution du voisinage. Cette technique s'apparente à une descente de plus forte pente détaillée dans le prochain chapitre (chapitre 9, page 107). Mais il existe d'autres techniques pour déterminer le successeur que nous détaillerons dans le paragraphe 8.5.1.

8.2.2 Voisinage

La notion de transition avec attribut permet en réalité de définir la notion de **voisinage**. Le parcours de l'espace de recherche se faisant de manière locale, on choisit (ou calcule) le successeur x^+ de la solution courante x^c parmi les solutions voisines $V(x^c)$. L'ensemble $V(x^c)$ est composé de l'ensemble des solutions que l'on peut atteindre par l'intermédiaire d'une transition depuis x^c .

8.2.3 Les attributs tabous

La recherche tabou doit son nom à la notion d'**attribut tabou**. Chacun des attributs associés aux transitions définies précédemment peut être positionné à l'état tabou à n'importe quelle itération de l'algorithme. Toute transition Tr_i possédant un attribut tabou sera elle aussi dite tabou. Pour une solution x^c , l'application de la transition tabou Tr_i conduit en une solution x_i elle aussi tabou. Le positionnement des attributs à l'état tabou permet de réduire le voisinage $V(x^c)$ en un voisinage $V(x^c, H) \subset V(x^c)$, en tenant compte de l'histoire H de la recherche ($x_i \in V(x^c)$ mais $x_i \notin V(x^c, H)$). Une première application de ces attributs tabous permet d'éviter les cycles dans la recherche (cf paragraphe 8.3.2). On s'interdit donc ainsi un retour immédiat en arrière (à la solution précédente).

8.3 La mémoire

8.3.1 Principe

La Recherche Tabou fait partie des **algorithmes à mémoire**, c'est à dire qu'à chaque itération la recherche d'une solution x^+ dans le voisinage $V(x^c)$ de la solution courante x^c est guidée par l'histoire de la recherche H .

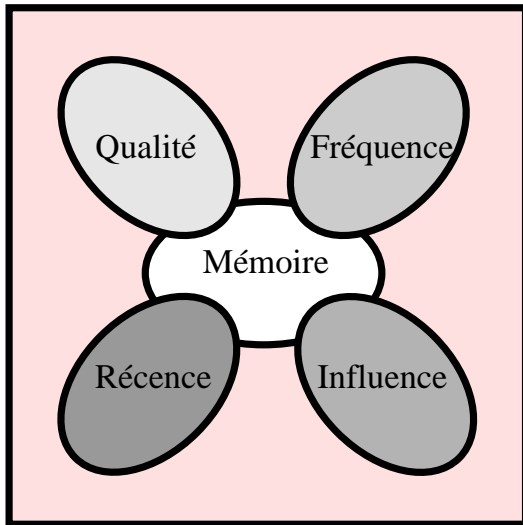


FIG. 8.1 – *Les structures de mémoire*

Dans l'ouvrage de F. Glover ([46]), cette histoire est analysée de quatre manières, chacune donnant lieu à une structure de mémoire différente. On distingue **la Qualité, la Fréquence, la Récence, l'Influence**.

Ces quatre structures peuvent être regroupées en deux types de mémoires : **la mémoire à court terme et la mémoire à long terme**.

La mémoire à court terme est composée uniquement de la récence des actions réalisées. La mémoire à long terme est quant à elle composée des trois structures restantes. Ces structures permettent de guider la recherche afin de répéter des choix performants et de ne pas perpétuer des erreurs commises.

8.3.2 La mémoire à court terme

La mémoire à court terme permet de construire le voisinage de la solution courante en fonction de l'histoire de la recherche. La structure mémorisant les dernières transitions utilisées (des derniers états traversés) permet de ne pas revenir sur des choix récents évitant ainsi les cycles sur des sous-ensembles restreints de solutions.

Cette mémoire utilise les attributs tabous pour interdire l'utilisation d'une transition depuis la solution courante. On retire ainsi du voisinage un ensemble de solutions déjà visitées.

Par exemple, si on a atteint la solution courante x^c en utilisant une transition $Tr : z \rightarrow x^c$ dont un attribut est $A_2 : z_i = 0, x_i^c = 1$ alors les transitions $Tr : x^c \rightarrow y$ dont un attribut est $A_3 : x_i^c = 1, y_i = 0$ sont tabous.

Un attribut est positionné à l'état tabou pendant un certain nombre d'itérations dépendant du problème traité. Ce nombre est à fixer avec précaution dans la mesure où il conditionne la taille des cycles. En outre, plus il est petit, plus on favorise l'intensification et plus il est grand, plus on favorise la diversification. On peut donc adapter la durée de maintien à l'état tabou en fonction de l'avancée de la recherche pour favoriser telle ou telle phase.

8.3.3 La mémoire à long terme

La mémoire à long terme peut prendre plusieurs formes. On introduit par exemple la notion de fréquence des transitions en repérant le nombre de fois qu'une transition (caractérisée par un ou plusieurs attributs) est utilisée. Conjointement on mémorise la qualité de la solution obtenue pour

savoir si la transition a permis d'améliorer le critère. Par comparaison des solutions de bonne (ou de mauvaise) qualité on peut en déduire leur structure afin de guider la recherche en construisant une partie de la solution. Enfin la mémorisation de l'influence de certains choix sur la structure des solutions permet d'accroître la robustesse de celles-ci.

Cette mémoire à long terme permet de guider la recherche de manière locale (à chaque itération, cf. paragraphe 8.4.1) mais également lors des changements de stratégie (cf. paragraphe 8.4.2).

8.4 Les phases de la recherche

8.4.1 Mise en oeuvre de l'intensification

L'intensification dans la recherche tabou consiste à continuer de guider la recherche dans le même sens. On favorise donc le type de transition le plus utilisé (notion de fréquence de la mémoire à long terme) en mettant à l'état tabou les transitions tendant à détruire la structure courante.

8.4.2 Mise en oeuvre de la diversification

La diversification peut être mise en oeuvre de plusieurs façons. De manière générale, on l'applique lorsque la recherche ne parvient pas à sortir du bassin d'un optimum local (mise en place de cycles, la solution ne progresse plus).

Lorsque l'on diversifie on peut souhaiter conserver la structure de la solution. Pour se faire on modifie uniquement le voisinage de la solution courante en changeant l'état tabou de certains attributs de transition.

Dans le cas où on ne souhaite pas conserver la structure de la solution, des changements sur la solution courante peuvent être opérés. Par exemple, on peut repartir d'une bonne solution déjà visitée précédemment (redémarrage).

De manière générale, la diversification consiste à rendre prioritaires les transitions les moins utilisées (notion de fréquence de la mémoire à long terme) afin de changer la manière d'explorer l'espace de recherche.

Dans sa mise en oeuvre la diversification peut conduire à dégrader le critère afin de sortir du bassin attractif d'un optimum local. Cette phase est tolérée par la méthode mais après un certain nombre d'itérations, on peut être amené à reconduire cette diversification.

8.5 Aspects avancés

8.5.1 La liste de candidats

Lorsque l'évaluation de toutes les solutions du voisinage n'est pas possible (taille trop importante ou temps de calcul trop long), on constitue une **liste de solutions candidates** (cf. Glover [39]), choisies dans le voisinage $V(x^c, H)$ de la solution courante. Le choix du successeur parmi la liste de candidats est ensuite réalisé de manière classique (choix du meilleur, tirage aléatoire non uniforme (cf paragraphe 8.5.3)).

Différentes stratégies sont proposées par Glover pour minimiser les calculs liés aux choix de la liste de candidats [46].

8.5.2 Les critères d'aspiration

Il est possible d'assouplir la notion d'attribut tabou en définissant des **critères d'aspiration**. Ces critères d'aspiration permettent d'utiliser une transition malgré son état tabou. Par exemple, si toutes les transitions sont tabous, on choisit de rendre non tabou la plus ancienne. On encore si il se trouve qu'une transition conduisant à une solution de bonne qualité est à l'état tabou (et si on a accès à cette information) à l'itération courante, alors on peut décider de changer cet état. On définit ainsi une liste de conditions de "repêchage" pour chaque transition (avec éventuellement des seuils d'acceptation pour ces conditions).

8.5.3 La recherche tabou probabiliste

Dans la recherche tabou, on ne laisse que très peu de marge au hasard tendant ainsi vers les méthodes déterministes. Il existe cependant une version **probabiliste** de la recherche tabou. Cette recherche consiste à évaluer chaque transition à la manière de la recherche tabou classique, c'est à dire en incluant l'état de la transition (tabou ou non), le gain apporté, et éventuellement une modification de la valeur en fonction de l'histoire de l'algorithme (pénalisation). Puis on associe un poids à chacune d'elles en fonction de son évaluation. On réalise ensuite un tirage qui tient compte de ces poids (plus une transition a une évaluation élevée, plus on a de chances de la sélectionner). La méthode peut être affinée en réalisant un tirage parmi les k meilleures solutions voisines de la solution courante, permettant ainsi de réaliser une évaluation grossière des solutions (on n'est pas sûr du classement relatif des solutions).

8.5.4 Les oscillations stratégiques

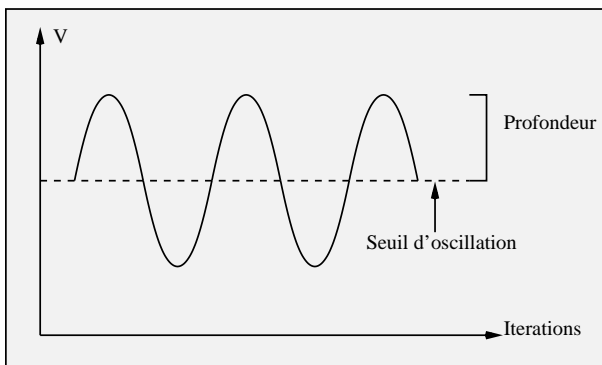


FIG. 8.2 – les oscillations stratégiques

On fixe ensuite une valeur seuil (S) à ce critère que l'on essaye de maintenir au cours de la recherche. Lorsque la valeur s'écarte trop du seuil, on change de phase afin de redresser l'évolution d'où une oscillation autour de la valeur seuil (cf. figure 8.2). On a donc une gestion dynamique des phases en fonction de l'état de la recherche (stabilité, convergence, ...).

Les oscillations stratégiques permettent de gérer l'alternance des stratégies. On peut par exemple gérer l'alternance d'utilisation de sous-groupes de transitions (constructives / destructives). On peut plus généralement encore, gérer l'alternance des phases d'intensification et de diversification au cours de la recherche. On définit pour cela un critère V qui va nous permettre de choisir quelle phase doit être appliquée. On fixe ensuite une valeur seuil (S) à ce critère que l'on essaye de maintenir au cours de la recherche. Lorsque la valeur s'écarte trop du seuil, on change de phase afin de redresser l'évolution d'où une oscillation autour de la valeur seuil (cf. figure 8.2). On a donc une gestion dynamique des phases en fonction de l'état de la recherche (stabilité, convergence, ...).

8.6 Algorithme de recherche tabou

Nous allons donner maintenant la structure d'un algorithme de recherche tabou probabiliste intégrant les différentes caractéristiques présentées et notamment les listes de candidats et les oscillations stratégiques.

Choisir une solution initiale x^c dans E
 $x^* \rightarrow x^c$
 $H \rightarrow \emptyset$ – l'histoire est vide initialement
Répéter
 Construire la liste de candidats $L(V(H, x^c))$.
 Déterminer x^+ parmi la liste de candidats.
 $x^c \rightarrow x^+$
 Si $c(x^c) < c(x^*)$ alors
 $x^* \rightarrow x^c$
 Fin si
 Mettre à jour H .
Jusqu'à critère de terminaison

Chapitre 9

La descente de plus grande pente

9.1 Introduction

Nous allons présenter ici la méthode de descente par recherche de plus grande pente. Cette méthode est une méthode analytique classique de recherche d'optimum.

Le problème que l'on souhaite résoudre ici est le problème dans sa formulation générale que l'on rappelle :

$$(P) \begin{cases} \text{Min}(f(x)) & f : R^n \rightarrow R \\ x \in R^n \end{cases}$$

La fonction f est par ailleurs considérée de classe C^1 (continuellement dérivable).

9.2 Méthode d'optimisation par direction de recherche

La méthode d'optimisation présentée ici procède par recherche de minimum selon une direction particulière dans l'espace de recherche. De manière simplifiée, l'itération k de l'algorithme est la suivante ([34]):

1. Déterminer la direction de recherche $d^{(k)}$.
2. Trouver $\alpha^{(k)}$ minimisant $f(x^{(k)} + \alpha d^{(k)})$.
3. $x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)} \cdot d^{(k)}$

9.2.1 Détermination de la direction de recherche

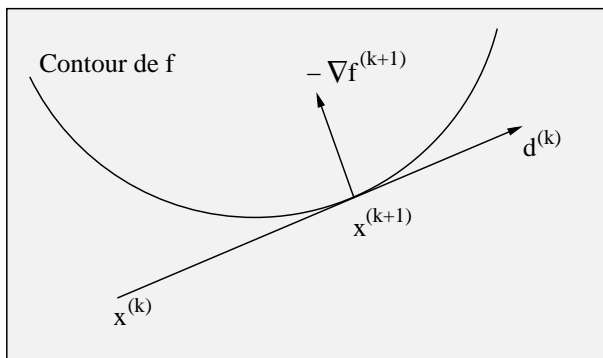


FIG. 9.1 – Direction de recherche

Les méthodes d'optimisation par direction de recherche diffèrent par la manière de choisir cette direction d . Cette étape peut être guidée par des connaissances sur le modèle du problème posé. De manière générale, la direction doit être choisie de telle sorte que : $(\nabla f(x^{(k+1)})|d^{(k)}) = 0$.

9.2.2 Détermination du pas de recherche

Après avoir déterminé la direction de la descente, il faut déterminer le pas α le long de cette direction de recherche.

L'algorithme présenté est idéalisé dans la mesure où l'on doit déterminer la valeur de α qui minimise $f(x^{(k)} + \alpha d^{(k)})$. Dans la pratique, cette étape est réalisée en discrétisant $f(x)$ (et éventuellement sa dérivée) pour différentes valeurs de $x = x^{(k)} + \alpha d^{(k)}$ le long de la direction de recherche.

9.2.3 Convergence

Sous les hypothèses de l'algorithme présenté précédemment, la méthode d'optimisation est globalement convergente car α minimise $f(x^{(k)} + \alpha d^{(k)})$ dans la direction $d^{(k)}$ ([21]).

Nous avons vu précédemment que dans la pratique la valeur de α n'est pas exacte. Sous de telles hypothèses la stabilité de l'algorithme n'est plus assurée.

9.3 La méthode de recherche de plus forte pente

Les méthodes de recherche par descente sont basées sur le principe de l'algorithme présenté précédemment.

9.3.1 Recherche de la direction

Pour ce cas particulier la direction de recherche est choisie tel que : $(\nabla f(x^{(k)})|d^{(k)}) < 0$ (f est supposée dérivable au point $x^{(k)}$).

Dans le cas de la recherche de plus forte pente, on pose $d^{(k)} = -\nabla f(x^{(k)})$. Cette direction est celle dans laquelle la fonction d'évaluation décroît le plus rapidement au voisinage de $x^{(k)}$.

Mais qu'en est-il de la méthode lorsqu'il n'est pas possible de disposer analytiquement de la dérivée de la fonction f ? Dans ce cas, les méthodes qui semblent être les meilleures ([34]) sont celles qui

estiment la valeur de la dérivé par une approximation en différence finie ([34]).

Si on considère que f est une fonction de \mathfrak{R}^n dans \mathfrak{R} ,

$$\nabla f_i(x^{(k)}) \simeq \frac{f(x^{(k)} + h.d_i) - f(x^{(k)})}{h}$$

avec $d_i \in \mathfrak{R}^n$ le vecteur n'ayant que sa i^{me} composante non nulle et égale à 1. ∇f_i est la dérivée partielle de f selon la i^{me} variable.

9.3.2 Détermination du pas de descente

Pour déterminer le pas de descente, on applique les règles de Walfe-Armija-Goldstein (WAG). Soit x^k la solution courante, on cherche x^{k+1} dans la direction d^k tel que :

1. **WAG 1** : $f(x^{k+1}) \leq f(x^k) + \sigma_1 \cdot \alpha (\nabla f(x^k) | d^k)$ avec $0 < \sigma_1 < 1$
2. **WAG 2** : $(\nabla f(x^{k+1}) | d^k) \geq \sigma_2 \cdot (\nabla f(x^k) | d^k)$ avec $0 < \sigma_1 < \sigma_2 < 1$

Si la fonction f est de classe C^1 et si elle est minorée alors on est assuré qu'il existe α tel que σ_1 et σ_2 existent.

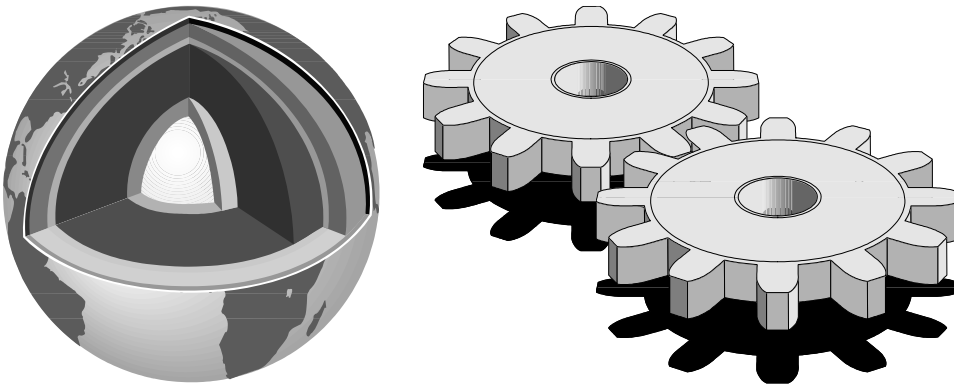
9.3.3 Convergence

La direction de recherche n'est pas exacte et l'objectif est plus une décroissance de la fonction d'évaluation qu'une recherche exacte de direction. Sous de telles hypothèses, la stabilité de l'algorithme n'est plus assurée.

Pour des raisons de rapidité de convergence, le test de convergence est réalisé en pratique de la manière suivante : si $|f(x^{(k+1)}) - f(x^{(k)})| \leq \epsilon$ alors on considère que l'algorithme a convergé.

Partie IV

Modélisation et résolution



Le contexte de notre étude étant fixé, les critères étant détaillés, nous pouvons maintenant nous concentrer sur la modélisation et la résolution du problème. En nous basant sur les méthodes d'optimisation décrites précédemment, nous allons présenter au cours du premier chapitre (chapitre 10) l'état de l'art des méthodes en optimisation de constellations de satellites. Nous présenterons dans ce cadre des méthodes issues du design de constellations en détaillant particulièrement la méthode dite de Hanson dont nous avons réalisé une implantation et une adaptation. Cette étude nous a apporté un certain nombre d'enseignements, dont nous tirerons parti pour la mise en place de notre approche. Nous présenterons également, dans ce même chapitre, la manière dont sont utilisés les algorithmes génétiques pour le design des constellations. Au cours du chapitre suivant (chapitre 11), nous décrirons le cheminement qui nous a conduit vers la méthode de résolution que nous proposons, en faisant notamment un bilan sur les obstacles auxquels nous avons à faire face pour mener à bien cette étude. La modélisation du problème sera ensuite détaillée. Cette dernière s'appuie essentiellement sur une décomposition adroite de l'espace de recherche qui nous concerne. Enfin nous décrirons avec précision les méthodes de résolution adoptées au cours des chapitres 12 et 13. En conclusion de cette partie, nous intégrerons les différentes composantes de notre approche pour présenter l'algorithme de résolution.

Chapitre 10

État de l'art

10.1 Introduction

Avant de présenter notre proposition de résolution nous allons donner quelques éléments sur les méthodes d'optimisation utilisées pour le design de constellations de satellites.

Trois parties composent cette présentation :

- dans un premier temps (section 10.2), des méthodes constructives élaborées spécialement pour le design de constellations de satellites vont être présentées. Ces approches sont basées sur une simplification du problème dans la mesure où l'on n'explore qu'un certain type de constellations.
- puis nous détaillerons (section 10.3) la méthode de Hanson dont nous avons réalisé une implantation et une adaptation. Cette méthode de construction d'orbites va nous permettre de tirer un certain nombre d'enseignements sur l'utilité et l'efficacité des méthodes directes.
- enfin nous présenterons (section 10.4) la manière dont certaines méthodes d'optimisation (algorithmes génétiques, recherche aléatoire adaptative, ...) sont utilisées pour le design de constellations.

L'ensemble des techniques présentées au cours de ce chapitre nous permettra d'identifier un certain nombre de difficultés et d'éléments de résolution que nous synthétiserons dans le chapitre suivant.

10.2 Présentation des algorithmes dédiés au design

10.2.1 Principe

M. Bello Mora réalise un travail de synthèse remarquable dans son article intitulé *ORION - A Constellation Mission Analysis Tool* ([84]). Il présente en effet les méthodes classiques de design de constellations, c'est à dire les méthodes permettant de mettre en place des constellations possédants

des propriétés de symétrie pour un problème homogène sur toute la terre.

On y trouve notamment un certain nombre de méthodes directes telles que les méthodes de Walker, Draim, SOC ou encore Hanson. Ces méthodes sont des méthodes de résolution géométrique ([5]) permettant d'obtenir un résultat de manière rapide et déterministe.

Le point commun des différentes méthodes est qu'elles construisent la constellation en ajoutant des satellites de manière à assurer une continuité du service au dessus des zones couvertes par les précédents. Pour mettre en place une telle approche, les satellites ajoutés doivent avoir le même comportement que les précédents, autrement dit l'altitude et l'inclinaison doivent être identiques et ces méthodes de construction conduisent naturellement à des constellations homogènes.

10.2.2 Exemple : la méthode SOC

La méthode *SOC* (Street Of Coverage) ([84]) que l'on pourrait traduire par *méthode des bandes de couverture* fut introduite par Lüders en 1961 ([82]). C'est une méthode itérative au cours de laquelle on ajoute (à chaque itération) un satellite à la constellation de manière à assurer la continuité (dans le temps) du service rendu. La figure 10.1 illustre ce mécanisme :

- on part d'une constellation composée du satellite 1 placé à une altitude h et une inclinaison i données. L'orbite est choisie de manière à ce que la trace du satellite se répète après un certain nombre de révolutions.
- On place ensuite, sur la même orbite, un second satellite de manière à ce que les deux satellites aient une zone en visibilité commune.
- On réitère le processus jusqu'à ce que l'orbite soit saturée (le n^{ime} satellite ajouté a une zone de visibilité commune avec le $(n - 1)^{ime}$ et avec le premier).

Ce procédé permet de construire la première bande de couverture continue (figure 10.1).

Après avoir assuré cette continuité au sein d'une bande, on généralise aux bandes voisines (figure 10.2). Un placement judicieux des deux bandes permet d'élargir la bande totale de continuité (bande de couverture inter plan).

10.2.3 Exemple : la méthode de Walker

Au même titre que la méthode *SOC*, la méthode de Walker ([100]) considère des constellations d'orbite circulaire dont l'altitude et l'inclinaison sont communes à tous les satellites. Il base sa construction sur un ensemble de plans comportant chacun un même nombre de satellites. Mais au lieu de s'intéresser à la continuité de couverture au sein d'un même plan, Walker montre que les trous de couverture apparaissent, en premier, au centre du cercle circonscrit à trois points sous satellites (deux satellites pris dans un plan et le troisième dans le plan adjacent) (figure 10.3). Il définit ainsi une distance d_{max} ([15]) entre les satellites pour éviter l'apparition de trous de visibilité

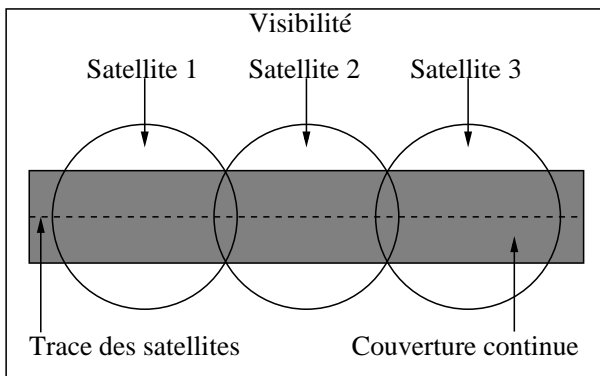


FIG. 10.1 – Agencement des satellites d'un plan par la méthode SOC

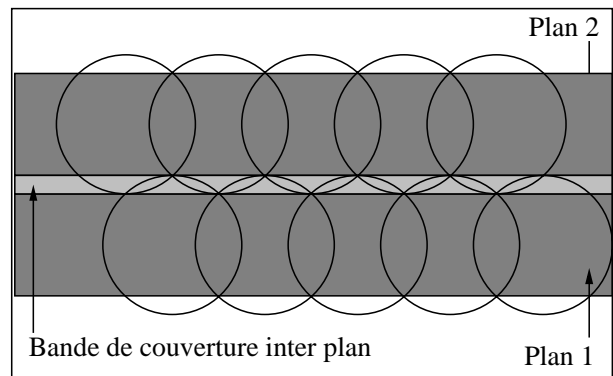


FIG. 10.2 – Agencement des plans par la méthode SOC

(figure 10.4).

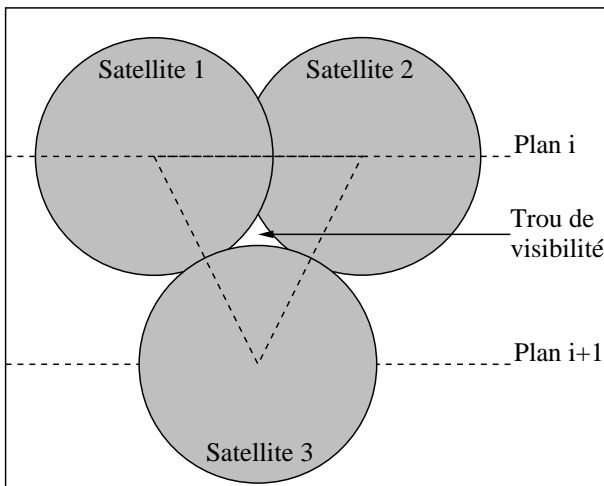


FIG. 10.3 – Apparition d'un trou de visibilité

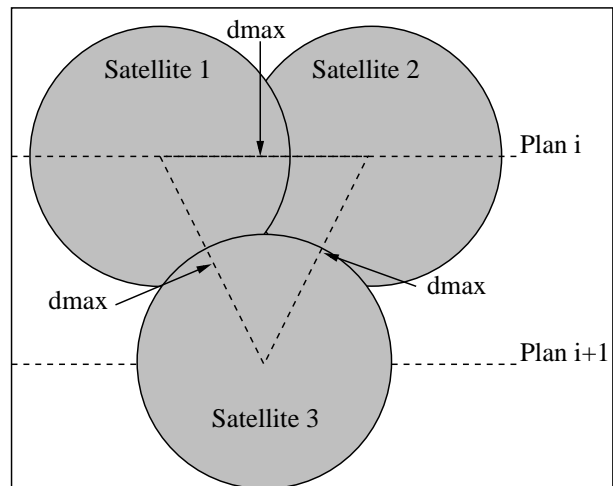


FIG. 10.4 – Trio de satellites de Walker

Définition des constellations de Walker

L'algorithme précédent permet de définir une véritable famille de constellations possédant des propriétés de symétrie. Cette famille de constellation fut définie en 1970 par J. G. Walker [100]. Il propose également un certain nombre de méthodes pour choisir la plus adaptée parmi l'ensemble de ces constellations.

On considère une constellation de n satellites définis par 6 paramètres orbitaux. Walker se place

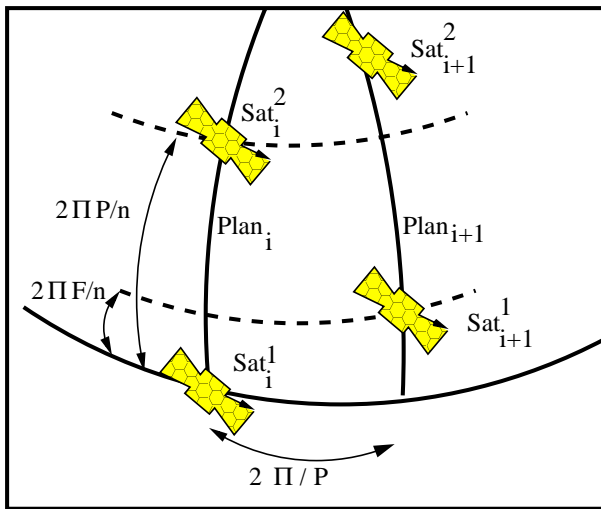


FIG. 10.5 – Les paramètres de Walker

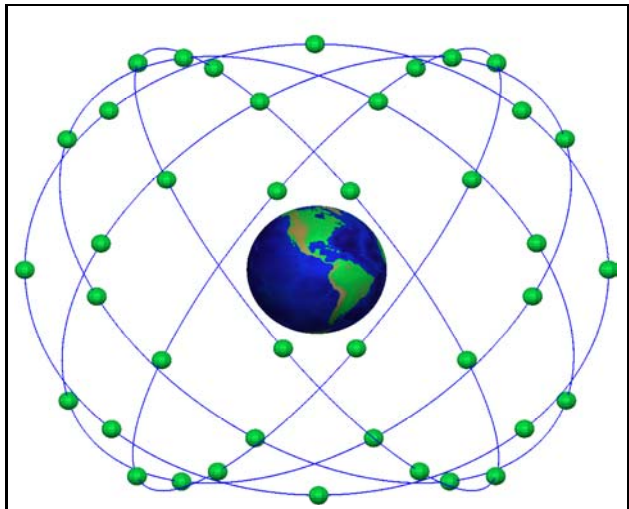


FIG. 10.6 – Constellation de Walker 40/5/3

dans les conditions suivantes pour définir ces constellations :

- Les orbites des satellites sont circulaires (pas de paramètre d'excentricité e , ni d'argument du périégée ω), on passe donc de $6n$ paramètres à $4n$.
- On considère des orbites de même demi grand axe a : on passe donc de $4n$ à $3n + 1$ paramètres.
- On considère des orbites de même inclinaison i : on passe donc de $3n + 1$ à $2n + 2$ paramètres.

Par ailleurs, il considère qu'une constellation est composée de P plans orbitaux répartis uniformément autour de l'équateur avec un écart angulaire de $2 * \frac{\pi}{P}$ (figure 10.5) entre deux plans consécutifs. On garde donc $2n + 2$ paramètres mais n d'entre eux varient dans l'ensemble de valeurs $\{1, \dots, P\}$. Enfin, les n satellites de la constellation sont répartis uniformément sur les plans : $\frac{n}{P}$ satellites par plan et un écart angulaire de $2 * \pi * \frac{F}{n}$ (figure 10.5) entre deux satellites consécutifs d'un même plan. Le nombre de paramètres est donc réduit à la donnée de n, P, a, i .

A ces quatre paramètres, Walker en ajoute un cinquième, appelé *phasage inter plan* et noté F , définissant l'écart angulaire entre le premier satellite d'un plan et le premier satellite d'un plan suivant. Cet écart angulaire a pour valeur $(2 * \pi * \frac{F}{n})$.

Les cinq paramètres de Walker sont donc n, P, F, a, i (figure 10.5) et un exemple de constellation de Walker 40/5/3 est donné figure 10.6. Ces constellations possèdent généralement de bonnes propriétés de couverture et Walker établit des tables permettant d'obtenir la plus petite constellation de Walker fournissant une couverture d'ordre quelconque. Typiquement les plans orbitaux sont inclinés entre 50 et 80 fournissant ainsi de bonnes performances sur les zones tempérées de l'hémisphère nord telles que les États-Unis et l'Europe.

10.2.4 Bilan

Nous pourrions ajouter aux deux méthodes présentées précédemment, d'autres méthodes telles que celle de A.H. Ballard ([5]) ou celle de D.C. Beste ([11]). Toutes ces méthodes ont la caractéristique commune d'être des méthodes de construction, géométriques, de constellation. Aucun processus d'optimisation n'est mis en oeuvre mais les hypothèses sur l'homogénéité de la constellation (répartition des satellites, répartition des plans) permettent d'obtenir une constellation de très bonne qualité (en termes de performance) et ce de manière déterministe.

Dans la suite de cette étude nous appellerons constellation symétrique une constellation définie, au même titre que les constellations de Walker, par un ensemble de plans équirépartis autour de l'équateur et comportant chacun un même nombre de satellites.

10.3 La méthode de Hanson

10.3.1 Introduction

A la différence des deux méthodes de construction présentées précédemment, qui se basent sur une couverture continue de la terre, la méthode de Hanson s'intéresse au problème de la couverture partielle de celle-ci. Cette formulation du problème et le caractère déterministe de son approche ont retenu toute notre attention.

Cette méthode, dédiée à des problèmes d'observation, est basée sur une minimisation du délai de revisite maximal (chapitre 4). Cet algorithme d'optimisation, mis au point par J. M. Hanson ([75], [63]), présente deux intérêts principaux. D'une part il permet de se convaincre encore un peu plus de l'efficacité des méthodes directes dans la recherche d'orbites (temps de calcul très faibles) et d'autre part de l'intérêt de l'analyse pointue de l'orbite proposée pour mesurer l'impact d'une modification des paramètres orbitaux.

Nous allons décrire à présent la méthode de Hanson appliquée à l'optimisation de l'orbite d'un satellite puis nous présenterons notre adaptation de la méthode dans le cadre de l'optimisation d'une constellation. Nous illustrerons ensuite la méthode avec quelques exemples de cas d'application. Enfin nous distinguerons les éléments de cette méthode qui nous ont été utiles dans l'élaboration de notre algorithme.

10.3.2 Principe de la méthode

La méthode de Hanson est une méthode directe permettant de définir de manière itérative l'orbite d'un satellite par la mise en oeuvre d'une méthode de construction. Elle a été développée au début des années 1990 lors de la montée de l'intérêt porté aux satellites de petite taille en altitude basse. Hanson s'est intéressé au problème de la couverture partielle du globe terrestre notamment

dans le domaine de l'observation ([75]).

Les critères utilisés par Hanson sont :

- le nombre de satellites qu'il cherche à minimiser pour réduire le coût du système,
- l'inclinaison des plans orbitaux qu'il cherche à minimiser pour maximiser la charge utile des lanceurs américains,
- les trous de visibilité qu'il cherche à minimiser pour assurer un certain délai de revisite,
- le temps de survol des zones d'intérêt qu'il cherche à maximiser pour augmenter le temps de service des satellites.

Les hypothèses sur les orbites étudiées sont :

- une inclinaison du plan orbital comprise entre 0 et 90 degrés (identique pour toutes les orbites de la constellation),
- pas d'orbite elliptique,
- des orbites à répétition de traces (phasées),
- un angle fixe d'élévation minimale pour les satellites.

Les paramètres variables de notre système sont donc l'altitude h du satellite, l'inclinaison i du plan orbital et la longitude du noeud ascendant Ω à l'instant 0. Nous allons présenter la théorie relative à la méthode dans le cas d'un unique point P situé sur la terre à une longitude $Long$ et une latitude Lat . On considère par ailleurs que P appartient à l'hémisphère nord et on souhaite maximiser le nombre de passages au dessus de celui-ci et/ou le temps total de survol.

La représentation qui est utilisée (figure 10.7) pour déterminer l'orbite optimale exprime la durée (ainsi que la plage horaire) pendant laquelle le satellite voit le point P en fonction de la longitude du point sous satellite à l'instant où celui-ci traverse l'équateur pour passer de l'hémisphère sud à l'hémisphère nord. La figure 10.8 représente des morceaux de trace au sol d'un satellite. Les parties conservées sont les parties appartenant à l'hémisphère nord dans une zone proche du point considéré. A travers cette représentation (figure 10.8) on voit que pour certaines valeur de Ω , la trace ne rencontre pas le cercle de visibilité centré sur le point P . Dans la représentation de la figure 10.7 cela se traduit par $\Omega < \Omega_{min}$ ou $\Omega > \Omega_{max}$. Pour deux valeurs particulières, la trace au sol coupe le cercle de visibilité en un unique point, ces valeurs de Ω (Ω_{min} et Ω_{max}) représentent les limites du lobe de la figure 10.7. On construit le lobe en balayant la plage de variation de Ω et en notant les instants de début et de fin de visibilité (T_1 et T_2 sur la figure 10.7). Dans l'exemple de la figure 10.7 :

- le satellite a une période de révolution T ,
- si le point sous satellite traverse l'équateur en Ω à l'instant t alors il sera en visibilité du point P sur la plage horaire $[t + T_1, t + T_2]$ au cours de la première période orbitale,
- pour $\Omega < \Omega_{min}$ et $\Omega > \Omega_{max}$, le satellite ne verra pas le point P pendant au moins une période orbitale.

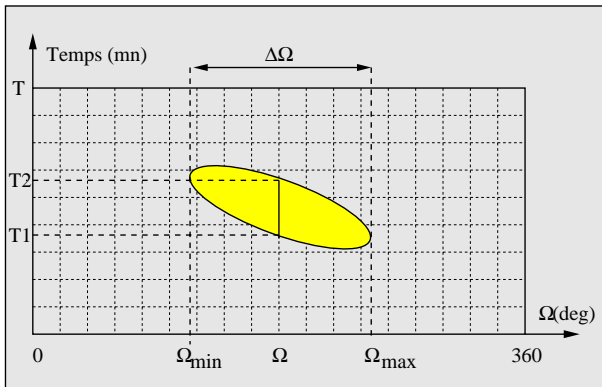


FIG. 10.7 – *Orbite à 1 lobe*

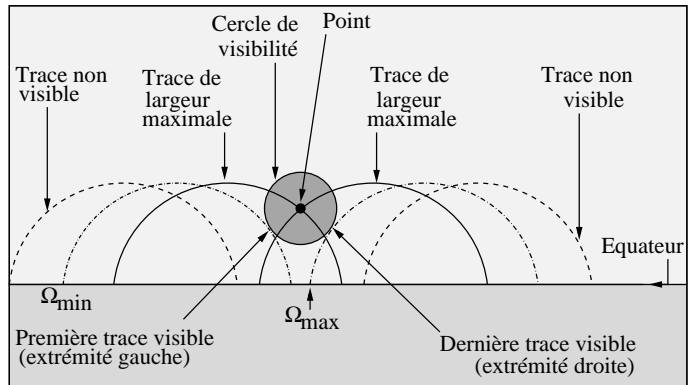


FIG. 10.8 – *Formation d'un seul lobe*

On peut également noter que pour certaines valeurs de l'inclinaison, il peut se former deux lobes (figures 10.10 et 10.9). Ces deux lobes apparaissent pour des valeurs élevées de l'inclinaison mais dont la valeur exacte dépend de la latitude du point P considéré. Les conditions pour qu'il se forme un ou deux lobes ainsi que les équations liant les divers paramètres sont décrites avec précision dans les documents [63] et [52].

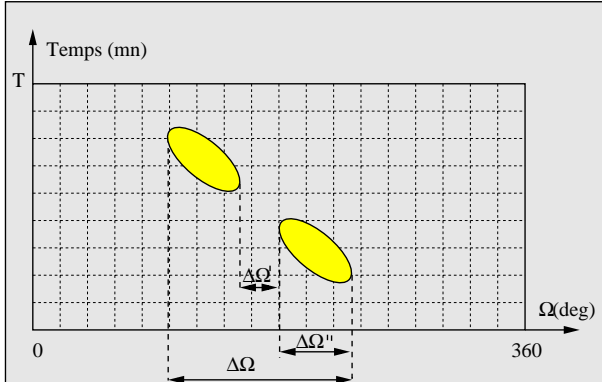


FIG. 10.9 – *Orbite à deux lobes*

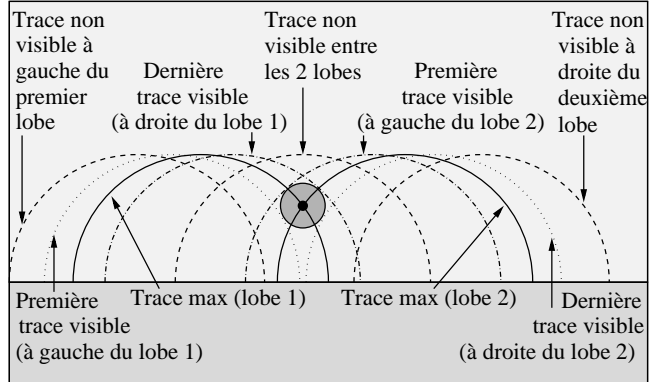


FIG. 10.10 – *Formation de deux lobes*

10.3.3 Nombre de passages au dessus du point d'intérêt

Lorsqu'un satellite commence à décrire son orbite en partant d'un point de longitude Ω_0 il retransverse l'équateur après un peu plus d'une période à une longitude différente Ω_1 . De même si le satellite fait une révolution de plus, il retransversera l'équateur à une longitude Ω_2 différente de Ω_1 avec la propriété suivante : $\Delta L = \Omega_1 - \Omega_0 = \Omega_2 - \Omega_1$. A altitude fixée on peut calculer la valeur

de ΔL , la valeur de l'angle d'élévation minimal et l'inclinaison donnent ensuite la forme du lobe. Il suffit alors de positionner de manière optimale (figures 10.11, 10.12) la longitude de départ du satellite pour que le nombre de passages au dessus du lobe soit maximal. Le nombre de passages

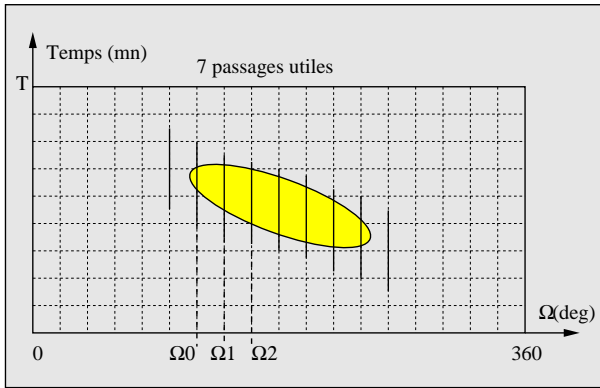


FIG. 10.11 – *Bon choix*

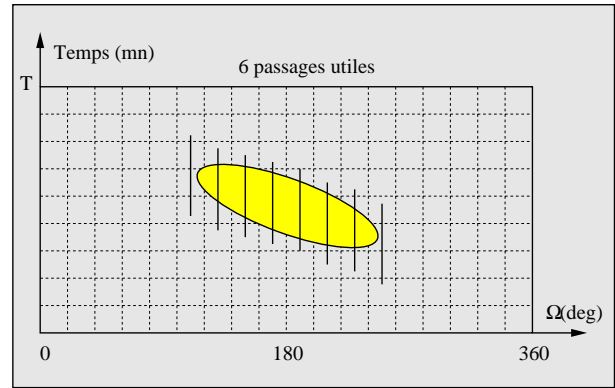


FIG. 10.12 – *Mauvaise choix*

utiles (en visibilité du point P) est déterminé par la largeur du lobe. Par exemple : si on considère un satellite d'altitude proche de 500km alors l'écart entre deux traces successives est de 24 degrés, si la largeur du lobe est supérieure à $7 * 24$ degrés alors il y aura 8 passages au dessus de la zone d'intérêt. On remarque également que les temps de passage les plus courts se situent aux extrémités du lobe, on aura donc intérêt, pour maximiser le temps de survol, à centrer les passages par rapport au lobe.

Il faut noter que dans le cas de deux lobes, des passages peuvent être perdus entre les deux lobes en fonction de leur écartement et de l'écart entre deux traces successives (figures 10.13, 10.14).

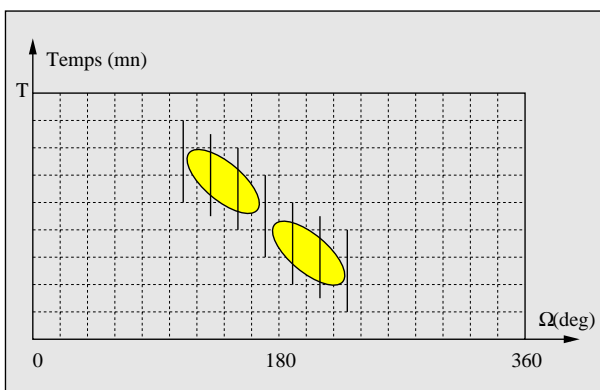


FIG. 10.13 – *Exemple 1*

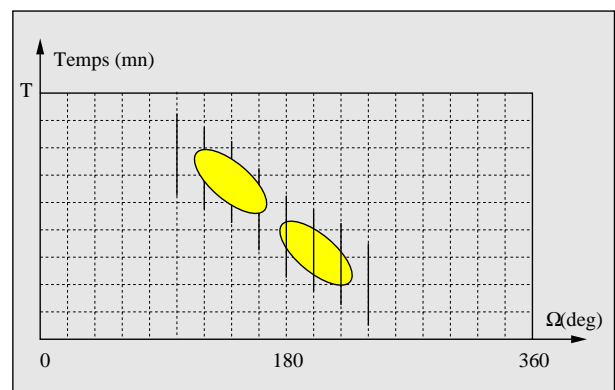


FIG. 10.14 – *Exemple 2*

10.3.4 Influence des paramètres i , h et Ω

La force de cette méthode réside dans la connaissance de l'influence des paramètres sur la valeur de la solution.

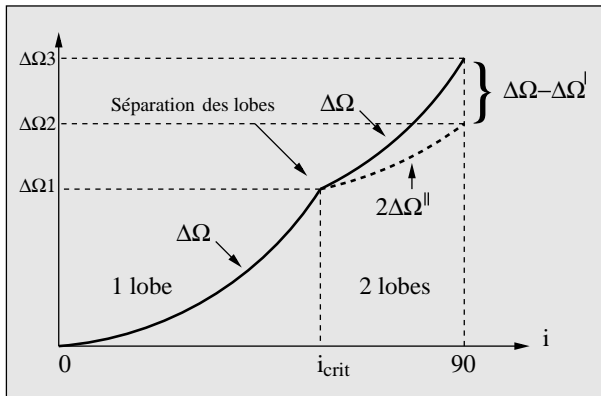


FIG. 10.15 – Influence de l'inclinaison

L'influence du paramètre h (altitude du satellite) (figure 10.16), se ressent au niveau de la durée de passage du satellite au dessus du point P (le lobe a tendance à s'arrondir) ainsi que sur la différence de longitude entre deux passages successifs du satellite.

La valeur de Ω n'influence pas la forme du lobe mais la position des traces par rapport aux lobes et donc le nombre de passages utiles.

Dans le cas où l'on n'a qu'un seul lobe, sa largeur ($\Delta\Omega$) augmente avec l'inclinaison jusqu'à ce que le lobe se sépare en deux.

Après la séparation, $\Delta\Omega$ continue de croître, ainsi que la largeur entre les lobes ($\Delta\Omega'$, figure 10.9), avec l'inclinaison, jusqu'à atteindre son maximum pour une inclinaison de 90 degrés. Parallèlement la largeur de chacun des lobes ($\Delta\Omega''$) diminue avec l'augmentation de l'inclinaison. La figure 10.15 récapitule l'évolution de la largeur des lobes en fonction de l'inclinaison. Le graphe est donné dans le cas général car quel que soit le point P et l'altitude h du satellite, sa forme ne change pas (seuls les valeurs i_{crit} , $\Delta\Omega_1$, $\Delta\Omega_2$ et $\Delta\Omega_3$ changent).

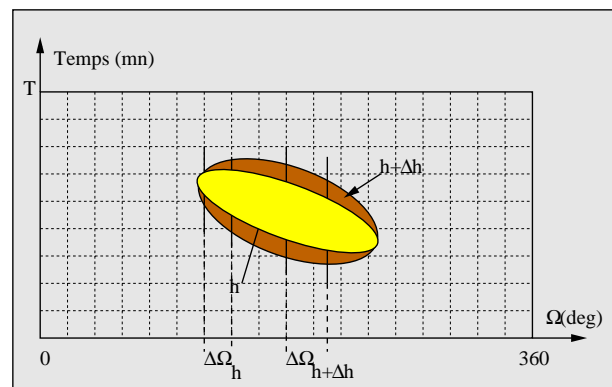


FIG. 10.16 – Influence de l'altitude

10.3.5 Adaptation et automatisation de la méthode

Tenant compte de toutes les remarques faites précédemment nous avons mis au point un processus qui réalise une recherche automatique (figure 10.17) de l'inclinaison i et du noeud ascendant Ω_0 maximisant le nombre de passages au dessus du (des) lobe (s). Le seul critère utilisé pour notre

optimisation est le nombre de passages au dessus du (des) lobe(s) que l'on souhaite maximiser. Étant donné que la largeur du lobe augmente avec l'inclinaison, nous avons intérêt à augmenter celle-ci pour augmenter le nombre de passages. Par contre, lorsque les lobes sont séparés la diminution de la largeur des lobes avec l'augmentation de l'inclinaison peut conduire à la disparition de passages utiles. De même, si les lobes sont séparés de plus de ΔL degrés, il y aura perte d'au moins un passage au dessus du point d'intérêt. Il faut donc déterminer l'inclinaison qui permet (pour une altitude fixée) d'avoir des lobes séparés par une bande de longitude $\Delta\Omega'$ légèrement inférieure à l'écart ΔL entre deux traces successives du satellite. Ainsi on aura le plus grand écart $\Delta\Omega$ entre les extrémités des lobes sans qu'il y ait perte d'un seul passage entre les deux lobes. Une recherche dichotomique est utilisée pour trouver rapidement l'inclinaison donnant cette séparation. La valeur de Ω_0 est ensuite calculée en fonction de la valeur de l'extrémité gauche du lobe gauche et la largeur $\Delta\Omega''$.

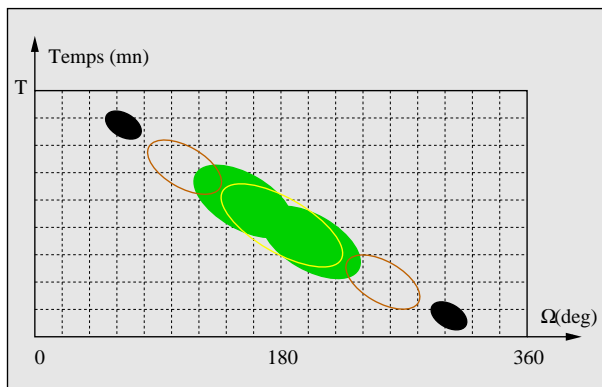


FIG. 10.17 – Recherche de l'inclinaison critique

10.3.6 Le critère

Dans le cas d'un seul satellite, les critères utilisés par Hanson se résument à minimiser l'inclinaison du satellite tout en augmentant le nombre de passages au dessus du point d'intérêt. Ces deux critères sont antagonistes puisque l'augmentation de la largeur du lobe (et donc du nombre de passages) est liée à l'augmentation de l'inclinaison. On aboutit par cette méthode à un compromis inclinaison / nombre de passages dépendant des poids attribués à chacun des critères.

Par ailleurs, le temps de survol n'étant pas identique à chaque passage, maximiser le nombre de passages ne veut pas nécessairement dire maximiser le temps total de survol. Dans beaucoup de cas on maximisera d'abord le nombre de passages et ensuite, par des variations faibles de la longitude du noeud ascendant, on maximisera le temps de survol. Mais on peut souhaiter maximiser le temps de survol des zones au détriment du nombre de passages. Le choix du critère dépend de l'application.

Par exemple, pour une mission de collecte de données de grand volume, le temps de survol de la zone à chaque passage doit être suffisant pour réaliser les transferts, et on préférera réduire le nombre de passages si cela contribue à l'augmentation des temps de survol; par contre, pour une mission de prise de vue, le temps de survol n'est pas un paramètre critique et on préférera augmenter le nombre de passages pour avoir plus de clichés de la même zone à des instants différents.

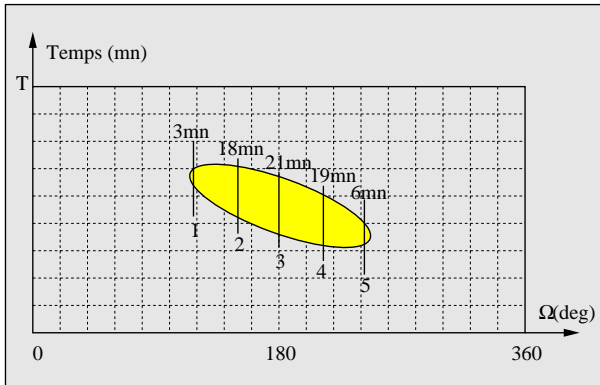


FIG. 10.18 – Temps de survol 58mn

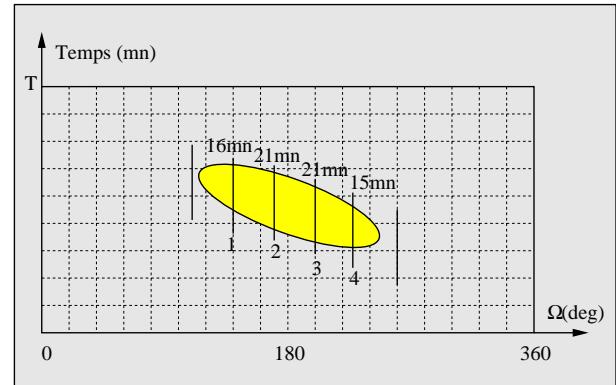


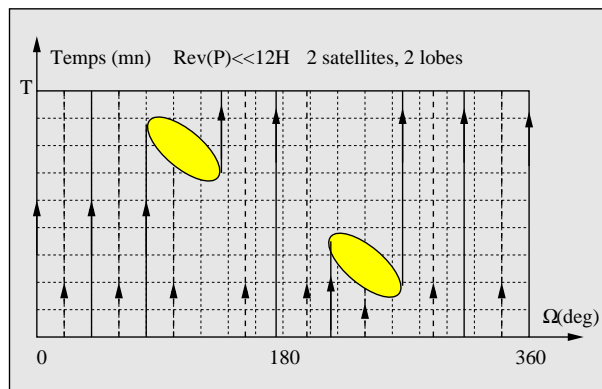
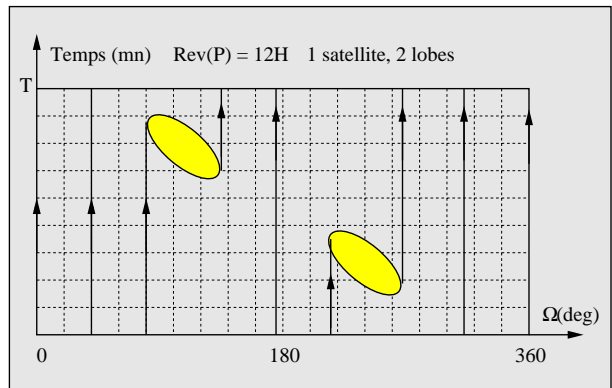
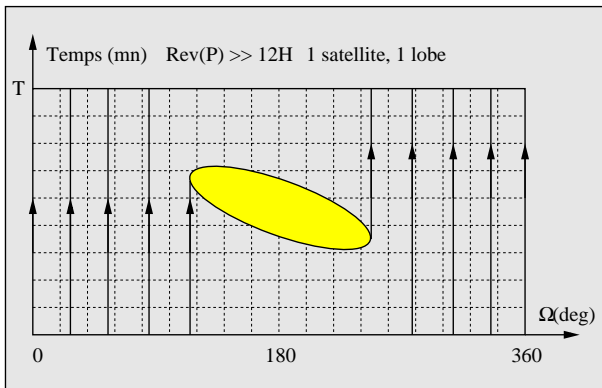
FIG. 10.19 – Temps de survol 1H13

La figure 10.18 présente un temps total de survol de $1H7mn$ ($3mn + 18mn + 21mn + 19mn + 6mn$) réalisé en 5 passages. Si on fixe un temps de survol minimal de $10min$ alors il n'y a que 3 passages utiles et le temps total descend à $58mn$. La figure 10.19 présente le même scénario mais avec une translation de la trace au sol permettant de passer à un temps de survol total de $1H13$ ($16mn + 21mn + 21mn + 15mn$) réalisé en 4 passages utiles.

10.3.7 Généralisation au design de constellations

On se place ici avec une contrainte supplémentaire qui est un délai de revisite maximal au dessus du point P . Lorsque ce délai est compris entre $12H$ et $24H$ (figure 10.20), le besoin peut être satisfait par un seul satellite et la méthode utilisée précédemment donne de très bons résultats en ayant un seul lobe de largeur maximale. Dans le cas où le délai de revisite est proche des 12 heures (figure 10.21), un seul satellite peut suffire mais on aura besoin cette fois de deux lobes pour satisfaire cette contrainte. En effet, dans le cas où l'on a un seul lobe, il se passe plus de 12 heures entre le dernier passage (vers l'extrémité droite du lobe) et le premier passage (vers l'extrémité gauche du lobe). Pour réduire ce temps de revisite, il faut séparer le lobe en deux afin de répartir les trous de visibilité et veiller à ce que le trou de visibilité entre la fin du premier lobe et le début du deuxième lobe soit inférieur au délai de revisite maximal. Il en est de même pour le trou de visibilité entre la fin du deuxième lobe et le début du premier. La recherche de l'inclinaison satisfaisant ces contraintes

peut conduire à réduire le nombre de passages utiles (réduction de la largeur des lobes) afin de mieux répartir les lobes. Dans le cas où le délai de revisite descend en dessous de 12 heures (figure 10.22), un seul satellite ne suffit plus. Il faut donc positionner un deuxième satellite afin de cadrer ses passages en fonction du précédent. Les passages utiles du deuxième satellite doivent se situer pendant le temps de revisite maximal du premier satellite. En fonction des contraintes imposées au système, il peut y avoir 1 ou 2 lobes.



10.3.8 Un nouveau problème

Le passage d'un satellite à plusieurs change la nature du problème et Hanson aborde ce problème de manière combinatoire : la difficulté n'est plus de trouver de manière directe les paramètres d'un satellite mais de placer les satellites les uns par rapport aux autres. La représentation change

également, et exprime la visibilité du satellite (depuis le point d'intérêt) en fonction du temps (figures 10.23, 10.24).

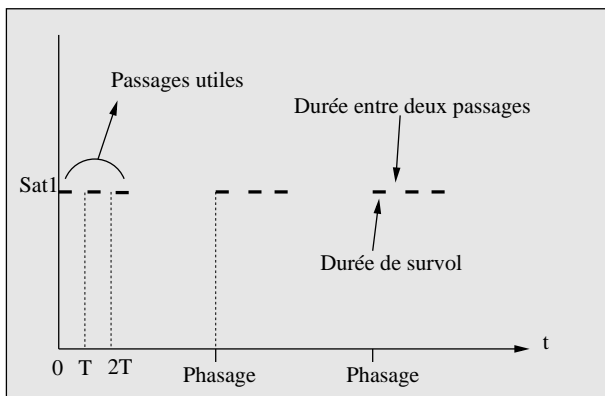


FIG. 10.23 – *Visibilité d'un satellite*

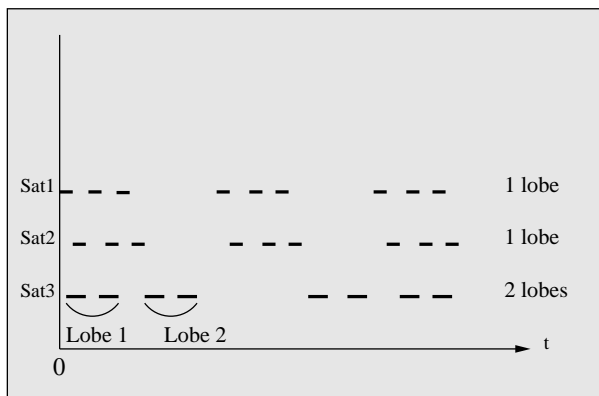


FIG. 10.24 – *Visibilité d'une constellation*

La méthode de placement devient un paramètre critique du fait de la forte combinatoire du problème surtout lorsque le nombre de satellites augmente (figure 10.25). Hanson nous indique dans son article ([63]) que pour 6 satellites les calculs s'étendent sur plusieurs semaines.

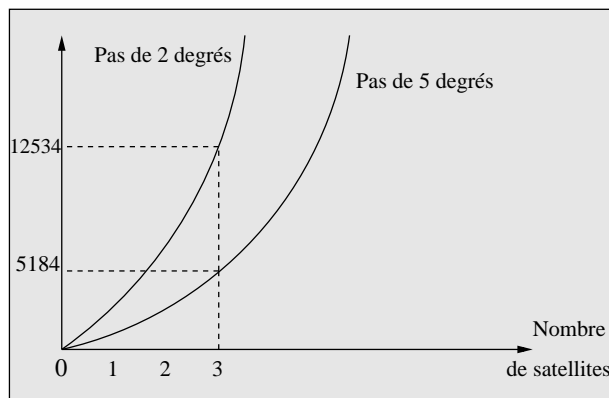


FIG. 10.25 – *Nombre de combinaisons*

Hanson donne les grandes idées pour résoudre ce problème combinatoire mais, à notre niveau, le but n'est pas de résoudre le problème de design de constellations à l'aide de la méthode de Hanson mais d'étudier les bénéfices d'une méthode directe de construction d'orbite. Nous avons donc décidé d'implanter un algorithme d'optimisation combinatoire très simple pour placer les satellites les uns par rapport aux autres. Ce processus est déterministe. Par ailleurs, la solution proposée sera la première solution acceptable et ne sera donc pas nécessairement optimale.

L'algorithme adopté est le suivant :

```
On applique la recherche dichotomique pour fixer
l'altitude, l'inclinaison et le noeud ascendant du
premier satellite de manière optimale (section 10.3.5).
tantque le trou de visibilité maximal est
    supérieur au délai maximal de revisite fixé
    Ajouter un satellite
    Placer le premier créneau de visibilité
    du nouveau satellite au milieu du trou de
    visibilité maximal
fin tantque
```

TAB. 10.1 – *Algorithme combinatoire de recherche de constellation*

En fin de processus, si le nombre de satellites atteint est trop important, on change l'ordre des trous de visibilité à combler et on relance le processus. Cette méthode présente l'intérêt d'être déterministe et de combinatoire très inférieure à une résolution du problème de manière globale et exacte (recherche de l'optimum global).

10.3.9 Généralisation au cas de plusieurs zones d'intérêt

On complique ici le problème précédent en ajoutant plusieurs points d'intérêt. Hanson propose deux approches pour résoudre le problème. La première méthode consiste à appliquer le même principe que précédemment en considérant une zone comme étant un unique point d'intérêt. Cette approche comporte deux inconvénients : tout d'abord les créneaux de visibilité se trouvent considérablement réduits du fait de la contrainte forte qui oblige le satellite à couvrir toute la zone ; d'autre part cette approche ne peut s'appliquer qu'à des zones relativement réduites pour ne pas conduire à une solution comportant trop de satellites. La deuxième méthode consiste à considérer chaque point séparément. Dans ce cas, la complexité augmente de manière significative et Hanson ne propose pas de méthode directe pour trouver la solution.

10.3.10 Application de notre algorithme

L'adaptation de cette méthode constitue un outil indépendant et autonome qui permet rapidement et de manière automatique de fixer un jeu de paramètres orbitaux pour un satellite en fonction

d'une zone restreinte à couvrir. Les critères utilisés (délai de revisite maximum, temps de survol minimum) orientent naturellement la méthode vers le domaine de l'observation.

Nous avons appliqué la méthode de Hanson dans le cas de satellites d'observation optique et radar. Ces deux cas diffèrent par la zone couverte par les instruments utilisés (chapitre 3, page 51). Des calculs spécifiques ont donc été envisagés.

Par ailleurs, l'aptitude de cette méthode à traiter des problèmes de navigation, où la position relative des satellites joue un rôle primordial, est loin d'être satisfaisante. Il en est de même pour des problèmes de télécommunications, où la couverture continue est requise : les performances de la méthode chutent de manière significative. Lorsque l'on souhaite déterminer une constellation fournissant une couverture continue des zones d'intérêt on pense naturellement à fixer un délai de revisite maximum égal à 0. Sous de telles hypothèses, l'espace de recherche devient plus important et les temps de calcul s'allongent surtout avec des orbites basses (plus de satellites nécessaires). La méthode comporte donc des limites assez franches pour le design d'une constellation.

10.3.11 Bilan

Cette étude sur la méthode de Hanson nous a conduit à dépasser les spécifications fournies par Hanson pour finalement intégrer et étendre (adaptation aux instruments radars) son approche dans un processus d'optimisation (recherche dichotomique, recherche combinatoire) élaboré par nos soins. Ce processus, bien que simpliste, permet de rendre compte de l'intérêt et de l'efficacité de cette méthode directe.

Cette méthode souligne de plus l'intérêt d'une analyse pointue de l'influence des variations des paramètres du satellite (altitude et inclinaison dans notre cas) sur le critère étudié.

Si cette méthode trouve assez vite ses limites dans la résolution de problèmes plus compliqués (augmentation de la zone d'intérêt, diminution du délai de revisite), elle n'en reste pas moins une source d'inspiration précieuse sur laquelle nous allons baser certains aspects de notre approche.

Dans l'approche de Hanson, chacun des satellites est évalué séparément, on peut ainsi se rendre compte de leur contribution au sein de la constellation. Si la constellation doit être considérée (dans le cadre de l'évaluation de ses performances) comme un ensemble de satellites et non comme n satellites indépendants (interaction et synchronisation des satellites), une étude séparée des éléments qui la composent nous a semblé digne d'intérêt et sera donc implantée dans notre algorithme.

10.4 Les algorithmes génétiques (AG)

10.4.1 Pourquoi les AG

Les algorithmes génétiques sont très utilisés en design de constellations ([69], [27], [94], [87]) car l'approche globale de la méthode, qui privilégie dans un premier temps l'exploration en répartissant

aléatoirement sur tout l'espace de recherche une population d'individus, permet de se défaire de connaissances sur la nature du critère (fonctionnement sous forme de boîte noire). La théorie nous assure effectivement que, statistiquement, au fur et à mesure des générations, les régions dignes d'intérêt vont être exploitées pour que la population converge vers l'optimum. On trouve ainsi dans la littérature ([67], [36], [88]) un grand nombre de cas d'application de ces méthodes.

10.4.2 Les limitations

Une stratégie simpliste en matière d'optimisation est bien souvent de considérer l'algorithme de résolution et le problème à résoudre de manière indépendante. On peut ainsi naïvement penser qu'un *AG* (parfois même déjà programmé dans une bibliothèque) peut résoudre n'importe quel problème d'optimisation mettant en jeu un critère dont on ignore tout. Cette application naïve des algorithmes génétiques au problème global de design de constellations ne conduit à rien. En effet, sans réaliser de spécialisations, l'application des différents opérateurs, définis pour des chaînes de bits, conduit à des aberrations structurelles. Par exemple, si on considère que l'on code une constellation par un chromosome binaire contenant la liste des $6n$ paramètres orbitaux, alors deux chromosomes différents peuvent coder la même constellation. Si on considère les deux individus suivants (Sat_1, Sat_2) et (Sat_2, Sat_1) (constellations composées de deux satellites) alors l'application d'un opérateur de croisement sur le site adéquat conduit à l'obtention des deux solutions stupides (Sat_1, Sat_1) et (Sat_2, Sat_2) .

Dans la pratique on ajoute à ce premier problème, un certain nombre de limitations dont T.A. Ely ([32]) donne un aperçu en énonçant que :

- d'une part la couverture de l'espace de recherche très vaste demande un nombre important d'individus pour se placer dans des conditions théoriques acceptables (et se rapprocher d'une convergence asymptotique).
- d'autre part le nombre d'évaluations étant très grand (beaucoup d'individus et beaucoup de générations) les temps de calcul sont prohibitifs.

Dans de telles conditions, l'application des algorithmes génétiques ne peut se faire que sur des sous problèmes (moins de paramètres, espace de recherche plus petit). C'est par exemple le cas de l'étude [36] qui propose une optimisation de la constellation GPS en ajoutant un satellite par plan orbital (on passe de 24 à 30 satellites en optimisant 6 paramètres).

10.4.3 Un opérateur de croisement inadapté

Si la simplification du problème permet de compenser l'inadaptation d'un codage naïf des chromosomes et la lourdeur de mise en oeuvre, l'opérateur de croisement reste souvent inadéquat sous sa forme standard : le croisement de deux bonnes constellations n'assure pas l'obtention d'une bonne

constellation.

Cela va même plus loin, puisque le croisement de deux jeux de paramètres viables ne conduit pas nécessairement à des constellations viables.

Pour retrouver des constellations viables après une application de l'opérateur de croisement, une fonction réparatrice est souvent mise en oeuvre ([68]). Avec un tel opérateur, l'algorithme génétique passe une grande partie de son temps à évaluer des constellations, à détruire les meilleures d'entre elles en les croisant pour ensuite réparer les dysfonctionnements.

Pour améliorer l'application des algorithmes génétiques, il faudrait coder un opérateur de croisement "intelligent", c'est à dire capable de déterminer les sites opportuns de croisement de manière dynamique (en fonction des parents sélectionnés).

La mise en place d'un tel opérateur nécessite d'élaborer des stratégies finalement proches de celles adoptées dans notre approche (chapitre 11).

Dans tous les cas, les opérateurs d'évolutions n'étant pas aisément contrôlables, on risque d'obtenir un algorithme génétique qui fonctionne comme une recherche aveugle et aléatoire.

10.4.4 Bilan

Si la théorie des algorithmes génétiques est alléchante, la prise en compte des paramètres réels tels que la taille de l'espace de recherche et le temps d'évaluation des solutions lors de leur mise en oeuvre ne permet pas de les appliquer correctement dans la résolution du problème global de design de constellations.

T.A. Ely ([32]) fait très justement remarquer que les algorithmes génétiques ont montré quelques limites dans le design de constellations de satellites et il n'applique pas directement ceux-ci à la résolution du problème. Les constellations sont construites avec une méthode appelée **Street Of Coverage** (figures 10.1 et 10.2, [32],[84]) et les algorithmes génétiques interviennent dans la recherche de la meilleure constellation construite avec cette méthode. L'intérêt de cette approche est que les constellations manipulées sont viables (pas de pénalisation à introduire et donc efficacité augmentée) par contre l'inconvénient est que l'on ne peut fournir que des constellations ayant la même structure.

10.5 Autres algorithmes évolutifs

Dans le panel des algorithmes évolutifs (dont font partie les algorithmes génétiques), certains auteurs préfèrent appliquer d'autres méthodes qui n'utilisent en fait que les opérateurs de sélection et de mutation dans des versions dérivées et spécialisées plus performantes.

Partant d'une solution (d'un ensemble de solutions), on cherche un (des) successeur(s) dans le voisinage de la solution en lui (leur) appliquant des perturbations. L'intérêt d'une telle approche

est que l'on abstrait le problème posé par l'opérateur de croisement : il n'y a pas de déstructuration des constellations.

Dans le cas d'une solution que l'on fait évoluer, cette version ressemble à une technique de recherche locale. Dans le cas de plusieurs solutions, on peut rapprocher la méthode d'une recherche locale dont on aurait lancé plusieurs instanciations en divers points.

Dans ce registre on trouve deux études ([7] et [9]) qui mettent en oeuvre une *Recherche aléatoire adaptative*. Dans le cadre de la première, il n'y a pas d'application de la méthode à la résolution du problème global. Il y a réduction du nombre de paramètres en considérant une constellation de 4 à 9 satellites en altitude géosynchrone. Dans la seconde étude, l'algorithme ne peut fonctionner que si on lui injecte une solution initiale. La solution trouvée en fin d'algorithme sera proche de la solution initiale en terme de structure. De plus il faut savoir dans quel sens orienter la recherche, donc avoir une idée de la solution cherchée. En conclusion, cet algorithme est surtout utilisé en seconde phase de design, c'est-à-dire après qu'une étude sur la forme et le dimensionnement de celle-ci a été réalisée.

Ces méthodes (tout autant que les algorithmes génétiques) sont appliquées sans spécialisation vis-à-vis du critère, toujours vu sous forme de boîte noire. Ceci permet de faciliter l'optimisation mais, même si la théorie indique qu'il est possible de trouver la solution optimale sous de telles hypothèses, dans la pratique, la complexité de l'espace rend la méthode trop aléatoire : on ne sait pas prédire l'impact d'une évolution sur le critère.

10.6 Conclusion

Nous avons abordé divers types de méthodes au cours de ce chapitre. Dans un premier temps nous avons vu que les méthodes constructives sont des méthodes déterministes bien adaptées au design de constellations symétriques (type Walker) dont la zone d'intérêt s'étend à la surface du globe (que ce soit en télécommunications ou en navigation) ([67],[88]).

Les algorithmes génétiques (et autres méthodes dérivées) sont quant à eux utilisés pour résoudre des sous-problèmes : nombre d'inconnues fixé, constellation partiellement définie.

Or l'émergence de nouveaux problèmes disymétriques (zones dispersées à la surface de la terre, plages temporelles non continues (cf chapitre 11)) remettent en cause l'optimalité de ces constellations dans le rapport entre la taille de la constellation et les performances délivrées. On s'oriente donc naturellement vers la recherche de nouveaux types de constellations non nécessairement symétriques mais fournissant un service plus localisé pour un coût plus faible.

Enfin, il faut noter que la manière de résoudre le problème est fondamentalement différent des approches proposées ici puisque l'algorithme qui sera proposé ne tient compte d'aucune restriction. Par exemple, ni la taille ni la structure de la constellation ne sont supposées être connues. Quelque soit

le problème que nous traiterons, nous appliquerons le même algorithme pour résoudre le problème. Seuls les paramètres de celui-ci pourront varier.

Chapitre 11

Modélisation : vers une nouvelle approche

11.1 Introduction

Avant de présenter notre travail, le lecteur est en droit de se demander pourquoi une nouvelle approche est nécessaire. On assiste d'une part à l'émergence de nouveaux besoins ([22]) dans le domaine de la navigation (GNSS, GLONASS [91]) et des télécommunications avec la nécessité d'intégrer des données multimédia (skybride, Iridium, ...) et l'apparition de zones d'intérêt plus restreintes. On s'intéressait avant à la totalité du globe et beaucoup d'études portent sur des couvertures continues de la surface terrestre ([24], [25], [26], [76], [77], [11]) conduisant à des constellations symétriques de type Walker. Par construction, les méthodes de Walker, *SOC* ou autre conduisent à des solutions symétriques. Les seules études sur des zones géographiques restreintes ont été réalisées sur des zones convexes ([90],[3], [23]) mais il existe pas ou peu d'études sur des constellations fournissant une couverture non continue de zones géographiquement éloignées. D'autre part, l'émergence de nouvelles constellations (constellations hybrides INES [18] composées d'une sous constellation LEO à 1500km d'altitude et de la constellation de 3 GEO GNSS1 [10]) met en avant la nécessité de spécifier un nouvel algorithme permettant de trouver des constellations non nécessairement symétriques (type Walker) afin d'accéder à une plus grande diversité des types de solutions envisagées.

Ce chapitre est consacré à la modélisation que nous avons adoptée et sur laquelle sera basé l'algorithme de résolution que nous proposerons au cours des chapitres suivants. Nous proposons cette approche pour résoudre le problème de design de constellations dans l'optique d'une recherche de constellations non nécessairement symétriques. Nous décrirons également au cours de ce chapitre le cheminement qui nous a conduit à la modélisation et à la méthode de résolution actuelle. Les différents éléments qui ont contribué à sa mise en place seront successivement rappelés. Nous fe-

rons également un bilan des faiblesses des algorithmes couramment utilisés ainsi que des techniques d'évaluation des constellations afin de présenter les points sur lesquels nous allons porter nos efforts. Enfin nous présenterons la structure de notre couple modélisation-résolution avant de décrire chacune des parties qui la composent.

11.2 Justification de notre approche

Les différentes parties qui constituent le contexte du design de constellations de satellites étant détaillées (à savoir le système étudié, la manière d'évaluer une solution et les approches existantes pour résoudre le problème) nous pouvons dresser un bilan sur les problèmes à résoudre. De manière générale nous allons essayer de fournir un élément de réponse à chacun des problèmes engendrés par les travaux existants et les contraintes intrinsèques au problème.

11.2.1 Un problème non standard

Dans un premier temps, l'étude des paramètres du système (chapitre 2) fait apparaître un nombre inconnu de paramètres (6 paramètres par satellite de la constellation, dimension de la constellation inconnue) et nous met d'emblée en présence d'un problème d'optimisation non standard que l'on ne pourra pas résoudre avec des méthodes classiques d'optimisation.

En outre, même à nombre de satellites fixé, les critères à minimiser sont trop irréguliers pour envisager une recherche de solution globale aux problèmes. La totalité de l'espace de recherche ne pouvant alors être explorée, des méthodes d'optimisation sous-optimales apparaissent donc comme privilégiées.

11.2.2 Un espace de recherche complexe

Par ailleurs, le nombre de satellites pouvant atteindre plusieurs dizaines voire plusieurs centaines (le nombre de paramètres est quant à lui six fois plus grand) nous rentrons dans la catégorie des problèmes d'optimisation de grande taille. Les méthodes sous-optimales requises doivent donc être basées sur des techniques permettant d'écarter efficacement certaines zones de l'espace de recherche. Dans ce cadre, nous proposerons dans la suite de ce chapitre (section 11.4) un modèle permettant de diviser l'espace de recherche. Nous avons vu (chapitre 3) qu'il existe un grand nombre de types d'orbites accessibles par un jeu de 6 paramètres orbitaux. C'est en partant de cette constatation que nous avons élaboré une classification des orbites permettant de décomposer l'espace de recherche. Chaque région de l'espace pourra ainsi être évaluée afin que l'algorithme de résolution utilisant le modèle puisse discerner les régions qu'il convient d'explorer et celles qu'il convient d'écarter.

En plus du nombre important de paramètres, la complexité est accrue par la nature hétérogène

de l'ensemble des paramètres (natures physiques et mathématiques différentes, chapitre 2). La présence de variables entières (nombre de satellites) et de variables réelles (les paramètres orbitaux) nous conduit à les prendre en compte de manière différenciée au niveau du processus d'optimisation. Néanmoins, si on considère que la constellation cherchée peut être composée de N_{min} à N_{max} satellites et compte tenu des applications et des dimensions du problème, résoudre exhaustivement (mais de manière sous-optimale) $N_{max} - N_{min}$ problèmes de design de constellations durant lesquels on considère le nombre de satellites fixé (possibilité d'appliquer des méthodes de résolution plus classiques), n'est pas satisfaisant.

Nous devons donc décomposer le traitement de ces paramètres en fonction de leur influence sur la constellation de manière à rechercher le nombre optimal de satellites ainsi que la valeur des paramètres orbitaux associés sans réaliser d'exploration exhaustive. La forte influence d'une variation du nombre de satellites sur l'évaluation de la constellation requiert un traitement particulier de ce paramètre. De la même manière, la différence d'influence des paramètres orbitaux sur l'orbite du satellite nous conduit à considérer deux groupes de paramètres, (a, i) et (e, ω, Ω, M) , avec des traitements, là encore, différents.

Cette hiérarchisation des paramètres nous conduit à envisager également une hiérarchisation de la recherche et donc la définition d'un algorithme de résolution à plusieurs niveaux.

11.2.3 Un critère coûteux et irrégulier

Une autre difficulté à laquelle nous sommes confrontés est la lourdeur de l'évaluation d'une constellation (chapitres 3, 4, 5). Le processus de simulation est en effet coûteux et nous allons faire un certain nombre d'efforts pour limiter les temps de calcul en simplifiant (localement et ponctuellement) le critère. Sous certaines hypothèses d'utilisation nous allons définir d'autres critères moins coûteux qui vont nous permettre d'accélérer la recherche.

Cette lourdeur de l'évaluation d'une solution nous amène également à exclure les algorithmes génétiques dans notre approche, ceux-ci conduisant à évaluer beaucoup d'individus à chaque génération. Pour dimensionner correctement la taille de la population des AG par rapport au nombre de paramètres à optimiser et à leurs natures, il faudrait une puissance de calcul trop importante et des temps de simulation inconcevables.

Nous allons donc faire le choix de considérer un algorithme d'inspiration locale mettant en oeuvre une unique solution afin de limiter les évaluations.

11.2.4 Une recherche pas trop aveugle

L'application des algorithmes génétiques avec un opérateur de croisement inadapté conduit trop souvent à la destruction des bonnes solutions sans pour cela conduire à de meilleures (chapitre 10). Ce problème provient du manque de connaissances sur les performances "individuelles" des satel-

lites composant la constellation. De cette constatation et de l'enseignement apporté par l'étude de Hanson, nous avons décidé d'analyser la constellation en profondeur lors de la simulation. Nous évaluerons par exemple chacun des satellites qui la compose afin de choisir les satellites qu'il faut supprimer ou remplacer. Cette analyse contribuera à augmenter le rapport entre le temps passé pendant la simulation et la quantité d'information qui en ressort (une unique valeur retournée par un critère d'évaluation sous forme de boîte noire).

Pour les mêmes raisons, le manque de connaissances sur l'influence, souvent imprévisible, des paramètres sur l'évaluation du critère, nous conduit à utiliser des algorithmes à mémoire capables de tirer un enseignement du parcours suivi.

11.3 Modélisation et résolution proposée

11.3.1 Principe de décomposition

L'idée directrice, indépendante des techniques utilisées pour résoudre le problème, est de guider efficacement la recherche au travers de l'espace défini par les constellations. Pour cela nous avons besoin de modéliser l'espace de recherche afin de le diviser et ainsi contrôler l'exploration des différentes régions.

Nous avons par ailleurs à traiter deux types de variables : le nombre de satellites (un entier n) et les paramètres orbitaux de ces satellites ($6n$ réels). Pour des raisons de discontinuité des critères associée à la variation du nombre de satellites et à la difficulté de mettre en oeuvre un algorithme capable de jongler avec des variables mixtes, nous avons décidé de scinder le problème en deux niveaux. Le niveau le plus bas a pour objectif la recherche des paramètres orbitaux "presque" optimaux à nombre de satellites fixé. Le niveau le plus haut a pour objectif de fixer le nombre de satellites de la constellation.

Mais la recherche des paramètres orbitaux au sein d'une région de l'espace devant être contrôlée, la modélisation donnera lieu à une abstraction de niveau intermédiaire.

Enfin, nous avons souligné au cours des chapitres précédents que l'évaluation d'une constellation était un processus coûteux. Pour pallier ce problème, nous allons fournir un effort pour décomposer l'évaluation d'une constellation et associer à chacun des niveaux définis une évaluation moins coûteuse.

La modélisation fera apparaître une décomposition de l'espace de recherche en trois niveaux :

- Au niveau le plus bas, les éléments manipulés sont des orbites afin de rechercher le jeu de paramètres orbitaux localement "optimal".
- Au niveau intermédiaire, on définit la notion de classes d'orbites (section 11.4) qui permet de définir les limites du niveau inférieur.

- Au niveau le plus haut, l'espace de recherche est composé de combinaisons de classes d'orbites (section 11.4) : on parle de configuration.

11.3.2 Principe de résolution

Les niveaux définis précédemment vont donner lieu à un algorithme combinant différentes techniques d'optimisation ([60]). Le niveau le plus haut fera appel à des techniques d'optimisation métaheuristiques. Les éléments manipulés par cette partie de l'algorithme sont des régions entières de l'espace des constellations dont les limites sont définies par la modélisation de l'espace de recherche. Ce premier niveau sera qualifié de *niveau métaheuristique (NM)*.

Le niveau le plus bas de l'algorithme sera quant à lui qualifié de *niveau numérique (NN)* et permettra une recherche très locale d'orbites optimales.

Le niveau intermédiaire, qui interface les deux précédents, sera lui qualifié de *niveau heuristique (NH)*. Comme nous le verrons dans la suite du chapitre, il représente la partie experte de l'algorithme et donnera lieu à une classification (potentiellement redondante et non exhaustive) conduisant à une véritable base de données orbitale (*ODB*).

11.3.3 Bilan modélisation-résolution

Avant de détailler les différents aspects introduits précédemment, voici un schéma synthétisant la décomposition du problème (figure 11.1) ([56]).

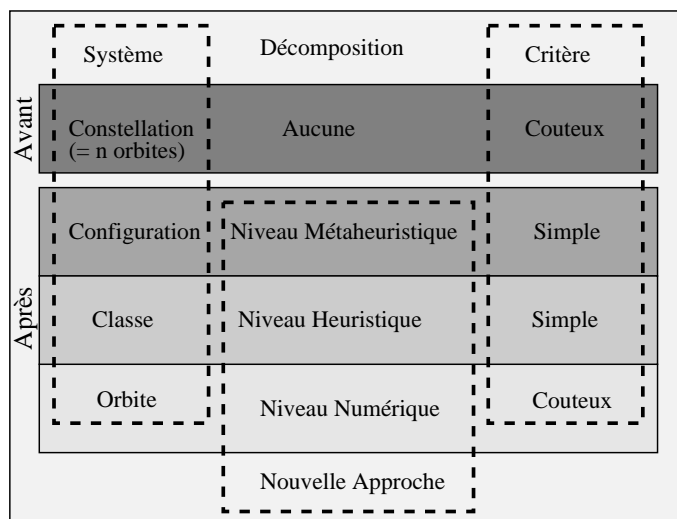


FIG. 11.1 – *Décomposition du problème*

Plusieurs notions sont présentes sur ce schéma. On y voit apparaître une décomposition du système qui passe de la représentation classique sous forme de constellations à la décomposition en trois niveaux :

- Le niveau bas de l’algorithme est donc le niveau numérique qui se base sur la notion d’orbite (formant une constellation) optimisée localement à partir d’une orbite initiale vis-à-vis du critère coûteux basé sur une simulation. Ce niveau tente de répondre à la question "*quelles orbites choisir?*".
- Le niveau heuristique introduit une modélisation de l’espace de recherche en définissant la notion de classes d’orbites. On tente de répondre à la question "*comment regrouper des orbites similaires et décomposer l’espace de recherche?*". Un critère simple d’évaluation est associé à ce niveau et permettra d’évaluer la compétence des différentes classes, face au problème posé. Un représentant d’une classe sera utilisé comme orbite initiale dans le niveau précédent.
- Le niveau métaheuristique se place au niveau le plus haut pour gérer l’intégration des différentes classes dans les configurations. Il tente de répondre à la question "*Combien de classes utiliser et lesquelles?*" en se basant sur une évaluation simple des configurations.

Nous allons maintenant présenter la modélisation et la décomposition de l’espace de recherche.

11.4 Décomposition de l’espace de recherche

11.4.1 Introduction

Nous allons décrire précisément dans le reste de ce chapitre la décomposition de l’espace de recherche. Cette modélisation représente la partie expertise de l’algorithme et est identifiée comme niveau heuristique de l’algorithme ([58]). Elle représente une véritable interface entre les deux niveaux de résolution du problème proposés (niveau numérique et niveau métaheuristique).

11.4.2 Notion de classe d’orbites

La grande diversité des traces au sol et la diversité des domaines d’application nous ont conduit à considérer que toutes les orbites ne sont pas équivalentes face à la résolution d’un problème. Pour traduire ces différences nous allons définir un modèle de classification des orbites qui sera utilisé pour constituer la base de donnée (*ODB*) dans laquelle l’algorithme (et notamment le niveau métaheuristique) puisera des orbites candidates afin de proposer des constellations.

La classification réalisée a pour but de regrouper des orbites ayant un certain nombre de propriétés connues. Ces propriétés peuvent être simples comme pour la classe regroupant les satellites d’orbite basse ($h \leq R_e + 1500km$) ou plus complexes comme celles définissant une orbite héliosynchrone.

Nous allons présenter la modélisation de ces deux types de classe avant de détailler un certain nombre de classes utilisées couramment.

Par ailleurs, nous verrons que l'ensemble des classes ne constitue par nécessairement une partition de l'espace de recherche. Nous verrons par la suite que cette classification distincte d'une partition (autorisant de la redondance aussi bien que de la non-exhaustivité), peut être un atout majeur lors de la mise en oeuvre de l'algorithme. **Nous conserverons, par abus de langage, le terme de classe d'orbites pour désigner un sous-ensemble des orbites possibles.**

11.4.3 Notion de configuration

La définition d'une classe d'orbites permet d'abstraire la notion de satellite. C'est-à-dire qu'une classe définit une plage de variation pour les paramètres orbitaux définissant l'orbite d'un satellite. De la même manière, on abstrait la notion de constellation et on définit la notion de configuration (section 11.10) qui n'est autre qu'une agrégation d'un certain nombre de représentants (éventuellement nul) de chacune des classes définies dans l'*ODB*.

11.5 Classes d'orbites définies par un intervalle

11.5.1 Principe

Nous allons présenter ici le premier type de classe d'orbites. Ce type de classe est défini par 6 intervalles (I) donnant la plage de variation de chacun des 6 paramètres orbitaux d'un satellite. Soit Cl_p une classe d'orbite, définie par $\{a_{min}^p, a_{max}^p, e_{min}^p, e_{max}^p, i_{min}^p, i_{max}^p, \omega_{min}^p, \omega_{max}^p, \Omega_{min}^p, \Omega_{max}^p, M_{min}^p, M_{max}^p\}$. Tout satellite Sat défini par les paramètres orbitaux $\{a, e, i, \omega, \Omega, M\}$ et tel que :

- $a \in I_a = [a_{min}^p, a_{max}^p]$
- $e \in I_e = [e_{min}^p, e_{max}^p]$
- $i \in I_i = [i_{min}^p, i_{max}^p]$
- $\omega \in I_\omega = [\omega_{min}^p, \omega_{max}^p]$
- $\Omega \in I_\Omega = [\Omega_{min}^p, \Omega_{max}^p]$
- $M \in I_M = [M_{min}^p, M_{max}^p]$

appartient à la classe Cl_p . Une classe peut donc contenir une infinité d'orbites, et donc de satellites.

11.5.2 Notion d'intervalle flou

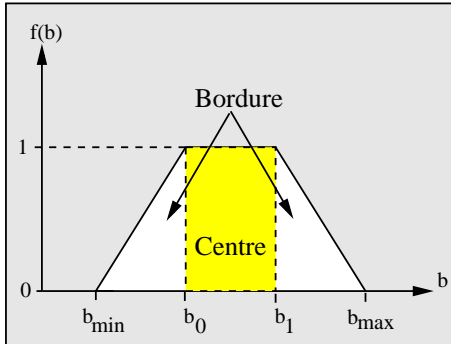


FIG. 11.2 – Intervalle flou

La notion d'arithmétique floue est introduite par Zadeh en 1965 ([102], [6], [28]) et se traduit ici, dans la définition d'un intervalle, par la donnée de 4 bornes $(b_{min}, b_0, b_1, b_{max})$, au lieu de 2 (b_{min}, b_{max}) , définies selon la figure 11.2). La fonction f évalue une mesure (une possibilité) d'appartenance à la classe : $f(b_i)$ exprime la possibilité pour que la variable b prenne la valeur b_i . Selon cette définition la figure 11.2 montre que la variable b est telle que :

- $b \geq b_{min}$ car $f(b) = 0 \forall b \leq b_{min}$
- $b \leq b_{max}$ car $f(b) = 0 \forall b \geq b_{max}$
- b a la plus forte possibilité de se trouver dans l'intervalle $[b_0, b_1]$ puisque $f(b) = 1 \forall b \in [b_0, b_1]$.

L'intervalle $If^K = [b_0, b_1]$ représente le centre de l'intervalle flou, et l'ensemble $If \setminus If^K = [b_{min}, b_0] \cup [b_1, b_{max}]$ sa bordure. Les éléments de la bordure ne sont pas les éléments les plus représentatifs de la classe mais ils appartiennent "presque entièrement" à celle-ci. En fonction du type d'orbite modélisé, sa représentation sous forme de classe donnera lieu à différentes formes mais toutes respectant la modélisation floue. Par exemple pour la classe géostationnaire, on a l'intervalle flou sur a qui est $\{a_{min} = Re + 36000km, a_0 = Re + 36000km, a_1 = Re + 36000km, a_{max} = Re + 36000km\}$ puisque qu'il n'y a qu'une altitude géostationnaire. Par contre la plage définissant la même altitude pour la classe des orbites basses aura comme intervalle flou : $\{a_{min} = Re + 300km, a_0 = Re + 400km, a_1 = Re + 1500km, a_{max} = Re + 3500km\}$. Nous donnerons le détail de classes couramment utilisées dans la suite du chapitre.

La notion d'intervalle flou remplace donc celle d'intervalle et on définit une classe par la donnée de 6 intervalles flous $\{If_a, If_e, If_i, If_\omega, If_\Omega, If_M\}$.

Lorsque l'on cherche à définir un ensemble de classes en se basant sur les différents types d'orbites connues, on se rend compte que les limites de certaines classes ne sont pas définies précisément. Si on considère les orbites basses (orbites *LEO*), dont la plage d'altitude classique est comprise entre 400 et 1500km, que dire d'un satellite à l'altitude de 1600km : est il en orbite basse, en orbite moyenne ou ni l'un ni l'autre? Pour pallier ce manque de précision, nous avons décidé de représenter chacun des intervalles définissant une classe d'orbite par des **intervalles flous (If)** ([59]).

11.5.3 Centre et bordure de la classe

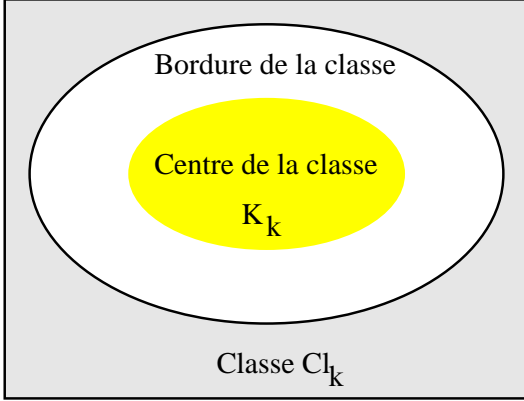


FIG. 11.3 – Représentation d'une classe

Le centre de la classe est donc le produit cartésien des centres des 6 intervalles flous définissant les paramètres orbitaux. Les éléments n'appartenant pas au centre de la classe ($Cl_p \setminus Cl_p^K$) définissent la bordure de celle-ci. Nous représenterons une classe comme dans la figure 11.3.

On définit la notion de centre de la classe Cl_p , noté Cl_p^K , l'ensemble des 6-uplets $\{a_p, e_p, i_p, \omega_p, \Omega_p, M_p\}$ tels que :

- $a \in If_a^K = [a_0^p, a_1^p]$
- $e \in If_e^K = [e_0^p, e_1^p]$
- $i \in If_i^K = [i_0^p, i_1^p]$
- $\omega \in If_\omega^K = [\omega_0^p, \omega_1^p]$
- $\Omega \in If_\Omega^K = [\Omega_0^p, \Omega_1^p]$
- $M \in If_M^K = [M_0^p, M_1^p]$

Le centre de la classe est donc le produit cartésien des centres des 6 intervalles flous définissant les paramètres orbitaux. Les éléments n'appartenant pas au centre de la classe ($Cl_p \setminus Cl_p^K$) définissent la bordure de celle-ci.

11.5.4 Égalité entre deux classes

L'égalité entre deux classes Cl_1 et Cl_2 est obtenue par l'égalité des intervalles flous qui la compose :

$$Cl_1 \equiv Cl_2 \Leftrightarrow \forall \alpha \in \{a, e, i, \omega, \Omega, M\} If_\alpha^1 = If_\alpha^2 \quad (11.1)$$

11.5.5 Distance entre deux classes

La distance entre deux classes Cl_1 et Cl_2 est définie par :

$$d(Cl_1, Cl_2) = \sum_{\alpha \in \{a, e, i, \omega, \Omega, M\}} \left| \frac{\alpha_0^1 + \alpha_1^1}{2} - \frac{\alpha_0^2 + \alpha_1^2}{2} \right| \quad (11.2)$$

11.6 Exemple de classes d'orbites définies par intervalle

Nous allons présenter ici les classes génériques que nous avons définies selon le premier type de classe (celle basée exclusivement sur la donnée de 6 intervalles flous). Nous donnerons la valeur des bornes de l'intervalle flou pour chacune de ces classes.

Les classes décrites permettent un recouvrement suffisamment important de l'espace de recherche mais cette "classification" n'est pas exhaustive et il existe une infinité de découpages possibles de cet espace. Pour définir ces classes nous nous basons sur une étude de la forme des traces au sol

des orbites ([55]). Nous allons présenter ces différentes traces au sol en étudiant les paramètres selon plusieurs points de vue et en décrivant leur influence avant de donner la valeur des bornes des intervalles flous.

Étant donné que seule la forme de la trace au sol nous intéresse pour cette présentation de la classification, nous allons uniquement étudier les paramètres (a, e, i, ω) . Les paramètres (Ω, M) ne modifient pas la forme de la trace au sol mais ont comme seule influence de translater celle-ci le long de l'équateur. Les plages de variations pour ces derniers paramètres seront prises comme maximales $(\{\Omega_{min} = 0, \Omega_0 = 0, \Omega_1 = 360, \Omega_{max} = 180\}, \{M_{min} = 0, M_0 = 0, M_1 = 360, M_{max} = 360\})$.

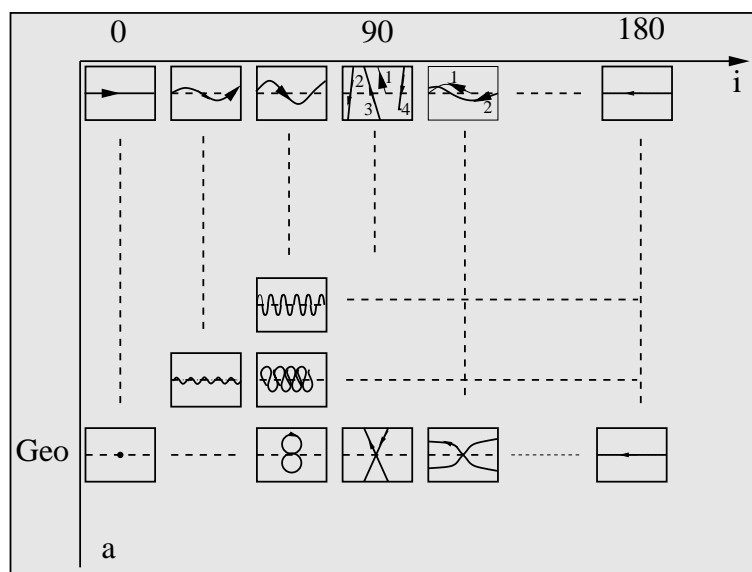


FIG. 11.4 – *Les orbites circulaires*

11.6.1 Étude des orbites circulaires

Nous simplifions ici le problème puisque l'orbite est circulaire (excentricité e nulle et argument du périégée ω sans influence sur la forme de l'orbite). Le domaine peut donc être représenté par un graphe en deux dimensions (figure 11.4). Cette figure permet de distinguer plusieurs formes de trace au sol dont nous allons décrire le comportement.

Les orbites basses

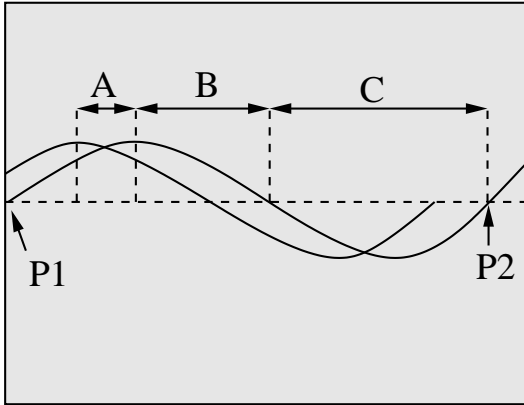


FIG. 11.5 – *Orbite basse*

- Plus a augmente, plus **A** augmente. Après un tour de la terre, la trace repasse donc plus loin de la première trace.
- Le paramètre **B** varie en fonction de l'excentricité (cf section 11.6.1).

La figure 11.5 présente la forme générale d'une trace au sol d'un satellite en orbite basse. Si on considère que le satellite commence sa révolution au point P_1 (le point sous satellite passe par l'équateur de l'hémisphère sud vers l'hémisphère nord), il aura réalisé une révolution au moment où il passera par le point sous satellite situé en P_2 . Le satellite aura parcouru toutes les longitudes lorsque le point sous satellite repassera par la longitude du point P_1 .

- Plus a augmente, plus **C** diminue. Il faut donc plus de révolutions du satellite pour faire un tour de la terre.

Étude des orbites à altitude géosynchrone

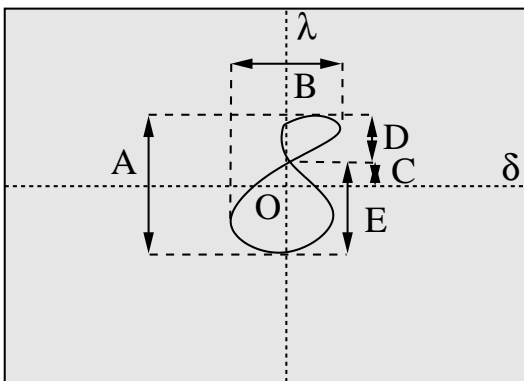


FIG. 11.6 – *Altitude géosynchrone*

d'orbites est un "huit" plus ou moins déformé en fonction de la valeur des trois paramètres restants. La figure 11.6 donne le schéma générique de la trace au sol de ce type d'orbites, on trouvera une visualisation plus précise des traces au sol avec les valeurs exactes des paramètres orbitaux dans le document [55].

Nous étudions ici les orbites à altitude géostationnaire. On rappelle qu'un satellite en orbite géostationnaire et d'inclinaison nulle paraît immobile dans le ciel de l'utilisateur (le satellite met exactement une journée pour faire une révolution d'orbite).

Dans le cas géosynchrone, l'altitude est la même que dans le cas géostationnaire mais l'inclinaison n'est pas nécessairement nulle. Le demi-grand axe (a) de l'orbite étant fixé, nous avons donc un espace de dimension trois à étudier (e, i, ω).

La forme générale de la trace au sol de ce type

Chacun des trois paramètres (e, i, ω) intervient de la manière suivante sur la trace au sol :

- Plus i augmente (jusqu'à 90), plus A augmente .
- Plus e augmente, plus B augmente.
- Plus e augmente, plus D diminue (C et E augmentent). Lorsque D devient nul, on dit que l'on a un point de rebroussement. La valeur de e pour laquelle on a un point de rebroussement dépend de celle de i .
- ω détermine la direction de l'axe du huit. Ce qui génère une symétrie de la trace ($\omega = 45$ et $\omega = 135$: symétrie par rapport à l'axe λ , $\omega = 45$ et $\omega = 225$: symétrie centrale, $\omega = 45$ et $\omega = 315$: symétrie d'axe δ).
- Pour $\omega = 0(\text{mod}180)$: $D = E = \frac{A}{2}$ et $C = 0$.
- Pour $\omega \neq 0(\text{mod}180)$: il y a dissymétrie et $c > 0$.

Les déformations possibles de la figure 11.6 sont nombreuses et quelques exemples sont donnés (figures 11.7, 11.8, 11.9, 11.10).

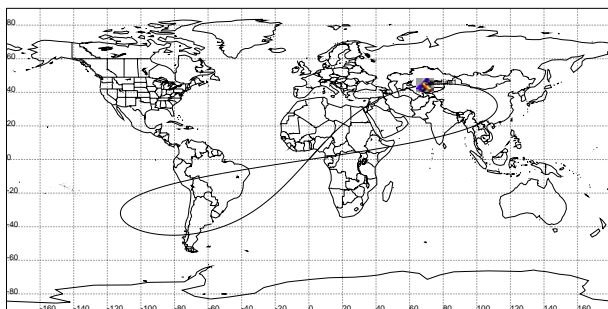


FIG. 11.7 - $e = 0.8, i = 45, \omega = 0$

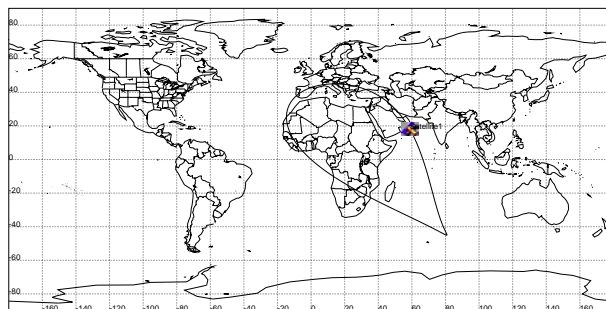


FIG. 11.9 - $e = 0.4, i = 45, \omega = 135$

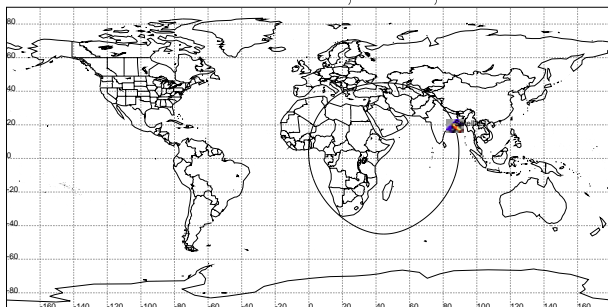


FIG. 11.8 - $e = 0.4, i = 45, \omega = 90$

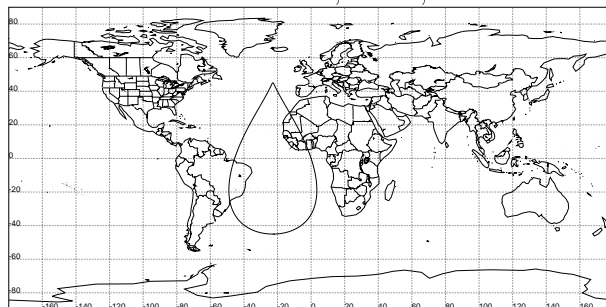


FIG. 11.10 - $e = 0.2, i = 45, \omega = 270$

11.6.2 Étude des orbites à haute-altitude

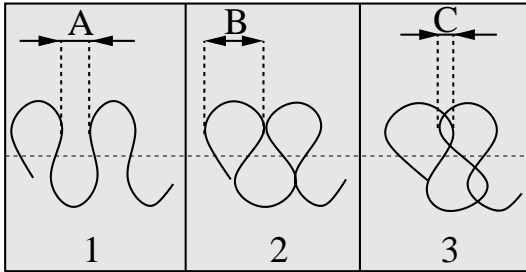


FIG. 11.11 – *Orbites haute altitude*

La vitesse des satellites à haute altitude se rapproche de celle de la terre. Aussi le nombre de révolutions nécessaires pour que le satellite fasse un tour complet de la terre augmente considérablement (un peu plus de 1 pour les basses altitudes, 2 pour l'altitude de 20000km et autour d'une dizaine pour des altitudes supérieures). A cette altitude, la forme de la trace au sol dépend conjointement de a et i .

Dans la figure 11.11 on étudie l'influence de a et

i sur le recouvrement de la trace.

- Figure 11.11.1 : pour a fixé, A diminue lorsque i augmente; pour i fixé, A diminue lorsque a augmente.
- Figure 11.11.2 : B diminue avec a .
- Figure 11.11.3 : pour a fixé, C augmente avec i ; pour i fixé, C augmente avec a .

11.6.3 Étude des orbites basses et des orbites moyennes excentriques

On suppose ici que l'altitude varie de 400km à 20000km . L'impact des paramètres orbitaux sur

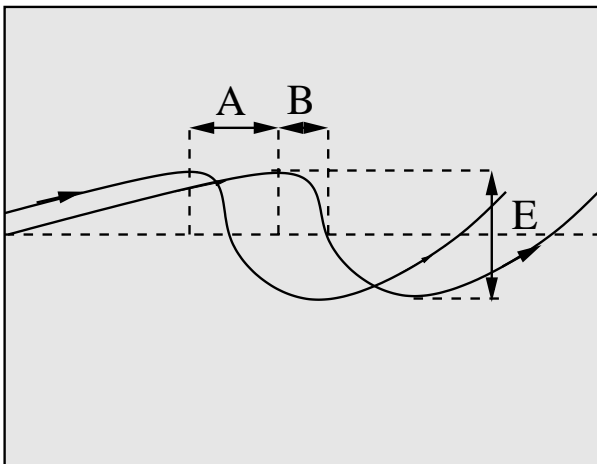


FIG. 11.12 – *Trace au sol avant le point de rebroussement*

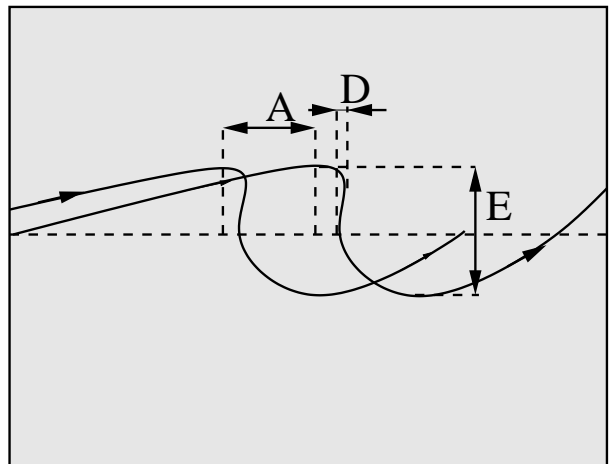


FIG. 11.13 – *Trace au sol après le point de rebroussement*

la forme de la trace au sol (figures 11.12 et 11.13) est le suivant :

- Quand e est nulle, on est dans le cas circulaire (cf avant).
- Quand e augmente, B diminue jusqu'à 0 (point de rebroussement).
- Quand e augmente après le point de rebroussement, D augmente.
- Quand i augmente, E augmente.
- A varie comme dans le cas circulaire.

11.7 Paramètres des classes génériques

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	400	400	36000	36000
e	0	0	0	0
$i(deg)$	0	0	180	180
$\omega(deg)$	0	0	0	0

Classe Circulaire

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	36000	36000	36000	36000
e	0	0	0.9	0.9
$i(deg)$	0	0	180	180
$\omega(deg)$	0	0	180	180

Classe Géosynchrone

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	400	600	1500	8000
e	0	0	0.9	0.9
$i(deg)$	0	0	180	180
$\omega(deg)$	0	0	360	360

Classe LEO

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	20183	20183	20183	20183
e	0	0	0	0
$i(deg)$	55	55	55	55
$\omega(deg)$	0	0	0	0

Classe GPS

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	6000	10000	20000	25000
e	0	0	0.9	0.9
$i(deg)$	0	0	180	180
$\omega(deg)$	0	0	360	360

Classe MEO

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	400	600	1500	8000
e	0	0	0.9	0.9
$i(deg)$	90	90	90	90
$\omega(deg)$	0	0	360	360

Classe Polaire

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	36000	36000	36000	36000
e	0	0	0	0
$i(deg)$	0	0	0	0
$\omega(deg)$	0	0	0	0

Classe Géostationnaire

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	400	600	1500	8000
e	0	0	0.9	0.9
$i(deg)$	80	85	95	100
$\omega(deg)$	0	0	360	360

Classe Quasi-Polaire

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	400	600	1500	8000
e	0	0	0.9	0.9
$i(deg)$	63.4	63.4	63.4	63.4
$\omega(deg)$	90	90	90	90

Classe Hémisphère Nord

α / borne	α_{min}	α_0	α_1	α_{max}
$h(km)$	400	600	1500	8000
e	0	0	0.9	0.9
$i(deg)$	63.4	63.4	63.4	63.4
$\omega(deg)$	90	90	90	90

Classe Hémisphère Sud

11.7.1 Conclusion

Cette étude des différents types de traces ainsi que de l'influence des paramètres sur leur forme, sera utilisée lors de la définition de classes plus précises que les classes génériques présentées. En effet, on peut par exemple souhaiter introduire dans la base de données une sous-classe de la classe géosynchrone. Cette classe pourrait par exemple regrouper des orbites géosynchrones dont l'excentricité est non nulle et l'apogée fixée au dessus de l'hémisphère nord. Les paramètres seraient par conséquent :

- Classe géosynchrone : $\{a_{min} = a_0 = a_1 = a_{max} = Re + 36000km, e_{min} = e_0 = 0, e_1 = e_{max} = 0.9, i_{min} = i_0 = 0, i_1 = i_{max} = 90, \omega_{min} = \omega_0 = 0, \omega_1 = \omega_{max} = 360, \Omega_{min} = \Omega_0 = 0, \Omega_1 = \Omega_{max} = 360, M_{min} = M_0 = 0, M_1 = M_{max} = 360\}$
- Classe géosynchrone dérivée : $\{a_{min} = a_0 = a_1 = a_{max} = Re + 36000km, e_{min} = e_0 = 0.2, e_1 = e_{max} = 0.4, i_{min} = i_0 = i_1 = i_{max} = 63.4, \omega_{min} = \omega_0 = \omega_1 = \omega_{max} = 270, \Omega_{min} = \Omega_0 = 0, \Omega_1 = \Omega_{max} = 360, M_{min} = M_0 = 0, M_1 = M_{max} = 360\}$

11.8 Classe définie par une relation entre des paramètres

11.8.1 Principe

Pour ce deuxième type de classes, les ensembles de variation des 6 paramètres orbitaux ne peuvent pas être définis séparément. Si on prend le cas d'une orbite phasée (c'est-à-dire telle que le satellite réalise $n + \frac{m}{q}$ révolutions par jour), il existe une relation complexe liant l'altitude du satellite à l'inclinaison du plan orbital. Si on souhaite définir une classe regroupant les orbites phasées à 2 jours (c'est-à-dire que le satellite repasse au dessus du même point après 2 jours), on obtient un ensemble de couples (a, i) discret, les autres paramètres quant à eux sont modélisés de la même manière que précédemment (par un intervalle flou). Dans ce registre on trouve également la classe héliosynchrone et toute autre classe définie pour un besoin particulier.

11.9 Odb : fonctionnalités

On regroupe l'ensemble des classes définies par intervalle ou par des relations dans une base de données orbitale notée ODB : $ODB = \{Cl_1, \dots, Cl_m\}$ avec $Cl_k = \{If_a^k, If_e^k, If_i^k, If_\omega^k, If_\Omega^k, If_M^k\}$ ($k \in [1, m]$) étant une classe d'orbite telle que $If_\alpha^k = \{\alpha_{min}, \alpha_0, \alpha_1, \alpha_{max}\}$ pour $\alpha \in \{a, e, i, \omega, \Omega, M\}$. Au niveau fonctionnel, on peut interroger l' ODB de plusieurs manières. En lui présentant un jeu de 6 paramètres orbitaux, l' ODB peut retourner :

- la liste des classes contenant ce jeu de paramètres (liste éventuellement vide).
- la classe la plus proche du jeu de paramètres.

L' ODB associe par ailleurs un représentant type à chacune des classes qui la composent.

11.10 Les configurations

11.10.1 Introduction

Comme nous l'avons déjà souligné, une infinité d'orbites peuvent appartenir à la même classe. Si dans le niveau numérique de l'algorithme la valeur précise des paramètres orbitaux de chacun des satellites est importante pour évaluer la constellation, ce n'est pas le cas au niveau heuristique où seule la classe d'appartenance du satellite est importante : La notion de classe abstrait donc celle de satellite au niveau heuristique.

De la même manière, nous allons remplacer la notion de constellation par celle de configuration. Si une constellation est la donnée de n satellites, une configuration est quant à elle la donnée d'un certain nombre de représentants de chacune des classes définies dans l' ODB .

Soit $\{Cl_1, \dots, Cl_m\}$ l'ensemble des classes présentes dans l' ODB . Une configuration Cf_p est définie comme suit : $Cf_p = \{Ns_1^p, \dots, Ns_m^p\}$, Ns_j^p étant le nombre de représentants de la classe Cl_j dans la configuration Cf_p .

Au niveau inférieur, chaque représentant d'une classe donne lieu à un satellite ayant un certain jeu de paramètres orbitaux satisfaisant les contraintes imposées par sa classe d'appartenance. La configuration se matérialise donc alors en une constellation. On constate qu'une configuration peut contenir une infinité de constellations. L'ensemble des constellations contenues dans la configuration Cf_p est l'ensemble :

$$\underbrace{Cl_1 \times \dots \times Cl_1}_{Ns_1^p} \times \dots \times \underbrace{Cl_m \times \dots \times Cl_m}_{Ns_m^p} = \{Cl_1\}^{Ns_1^p} \times \dots \times \{Cl_m\}^{Ns_m^p}.$$

De la même manière que pour les intervalles flous et les classes, on définit la notion de centre (Cf^K) et de bordure ($Cf \setminus Cf^K$) d'une configuration Cf comme étant le produit cartésien des centres des classes d'appartenance des satellites qui la composent : $Cf^K = \{Cl_1^K\}^{Ns_1} \times \dots \times \{Cl_m^K\}^{Ns_m}$. Nous représenterons une configuration de la même manière qu'une classe (figure 11.3)

11.10.2 Distance entre deux configurations

On définit la notion de distance $d(.,.)$ entre Cf_1 et Cf_2 par la relation :

$$d(Cf_1, Cf_2) = \sum_{i=1}^m |Ns_i^1 - Ns_i^2| \quad (11.3)$$

11.10.3 Relation d'équivalence

On définit la relation d'équivalence \equiv entre Cf_1 et Cf_2 par la relation :

$$Cf_1 \equiv Cf_2 \iff \forall i \in [1, m], Ns_i^1 = Ns_i^2 \quad (11.4)$$

11.10.4 Relation d'ordre

On définit la relation d'ordre \prec Cf_1 et Cf_2 par la relation :

$$Cf_1 \preceq Cf_2 \iff \forall i \in [1, m], Ns_i^1 \leq Ns_i^2 \quad (11.5)$$

si par ailleurs $\exists j \in [1, m] Ns_j^1 < Ns_j^2$ alors la relation est stricte (\prec).

11.10.5 Remarques

Avec la relation d'ordre précédente, on voit que toutes les configurations ne sont pas comparables. C'est le cas pour tout couple de configurations (Cf_1, Cf_2) tel que : $\exists i \in [1, m] Ns_i^1 < Ns_i^2$ et $\exists j \in [1, m] Ns_j^1 > Ns_j^2$.

D'autre part, deux configurations peuvent être équivalentes sans pour autant que les représentants des classes soient identiques deux à deux.

11.11 Quelques remarques

En définissant les classes d'orbites puis les configurations, nous sommes passés d'un espace de recherche dont chaque élément est une constellation, à un espace composé de configurations. L'avantage de ce nouveau découpage de l'espace est que celui-ci se trouve réduit et structuré ce qui facilite l'exploration. Par ailleurs, nous avons vu que deux classes d'orbites ne sont pas toujours disjointes : une constellation peut donc appartenir à plusieurs configurations. Inversement, selon la composition de l'*ODB*, une constellation peut n'appartenir à aucune des configurations engendrées.

Il n'existe pas une unique manière de décomposer l'espace de recherche et la définition des classes appartenant à l'*ODB* doit être réalisée avec soin avant toute résolution de problème. De cette bonne définition dépendent d'une part le temps de résolution du problème et d'autre part la nature des solutions trouvées. Toute donnée sur la nature des solutions recherchées (ou à écarter) doit être

prise en compte dans la définition de l'*ODB*. On essaye ainsi de restreindre la recherche de solutions parmi les caractéristiques souhaitées (orbite basse, orbite héliosynchrone, inclinaison polaire, ...).

Le recouvrement, total ou partiel, de certaines classes par d'autres joue aussi un rôle. Plus les classes sont disjointes, plus on évite de parcourir les mêmes régions, dans l'espace des constellations, à travers différentes configurations. Par contre le recouvrement de configurations permet, lors de l'application de la recherche de niveau supérieur (niveau métaheuristique chapitre 13), de sélectionner élégamment le successeur de la configuration courante (figure 11.14). Nous illustrerons

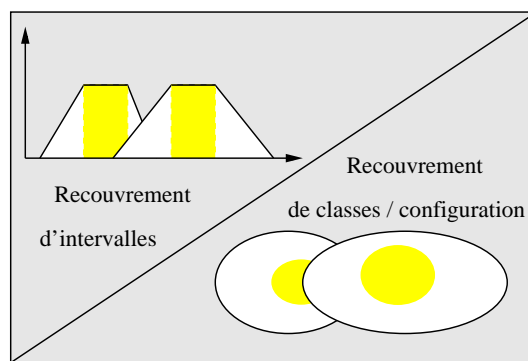


FIG. 11.14 – *Recouvrement de configurations*

l'influence du choix des classes au cours de la partie VI page 183.

11.12 Conclusion

Nous avons maintenant présenté une nouvelle étape dans la modélisation du problème : la décomposition de l'espace de recherche. Cette modélisation met en évidence le besoin de deux niveaux de résolution (niveau numérique et niveau métaheuristique) que nous allons détailler dans les chapitres suivants. Nous présenterons dans ce cadre les algorithmes de résolution adoptés en soulignant les efforts réalisés pour simplifier le critère et pour introduire de nouveaux moyens d'évaluer les solutions.

Chapitre 12

Résolution : le niveau numérique

12.1 Introduction

Le niveau numérique est le niveau bas de l'algorithme. Il est mis en oeuvre dans le but d'explorer une certaine région de l'espace de recherche délimitée par les niveaux heuristique de l'algorithme. C'est une phase d'intensification de la recherche dont l'objectif est de trouver la meilleure solution dans la région proposée.

L'algorithme de résolution proposé manipule des constellations dont les satellites suivent des orbites délimitées par les classes d'orbites définies au cours du chapitre précédent. Chaque satellite est en effet un représentant de l'une des m classes de l'*ODB*.

Nous nous intéressons ici uniquement à la recherche d'une solution sous-optimale au sein de la région fixée, mais nous verrons par ailleurs au cours du chapitre 13 que le niveau métaheuristique exploite cette recherche pour évaluer la région.

Les hypothèses faites dans ce chapitre sont :

- soit C_s une constellation composée de n satellites.
- Les 6 paramètres orbitaux des satellites $Sat^k, k \in [1, n]$ de la constellation C_s prennent leurs valeurs dans un intervalle restreint tel que : $\forall k \in [1, n]$

$$- a^k \in [a_{min}^k, a_{max}^k]$$

$$- e^k \in [e_{min}^k, e_{max}^k]$$

$$- i^k \in [i_{min}^k, i_{max}^k]$$

$$- \omega^k \in [\omega_{min}^k, \omega_{max}^k]$$

$$- \Omega^k \in [\Omega_{min}^k, \Omega_{max}^k]$$

$$- M^k \in [M_{min}^k, M_{max}^k]$$

- Le nombre n de satellites est fixe.
- On évalue la constellation C_s à l'aide du critère f basé sur une simulation (critère coûteux mais "exact", c'est-à-dire physiquement justifié).
- La valeur des paramètres de la constellation initiale C_{s_0} est fixée.

- Le voisinage de la constellation Cs est un ensemble de constellations noté $V(Cs)$.

Par ailleurs, le critère utilisé n'est pas un critère d'acceptation de la constellation dans le sens où il ne renvoie pas un calcul de pire cas de performance au cours de la simulation. Lorsque l'on doit valider une constellation qui doit satisfaire un certain seuil de performance sur une certaine zone et pendant une certaine période, on calcule souvent la pire des performances sur la simulation pour savoir si la constellation satisfait ou non les contraintes fixées. Nous allons préférer une évaluation qui rend compte des performances moyennes durant la simulation afin d'éviter une évaluation binaire des constellations.

Il faut noter enfin que l'espace de recherche, même restreint aux intervalles décrits précédemment, reste trop grand pour envisager une recherche exhaustive.

12.2 Principe

Sous les hypothèses décrites précédemment (section 12.1), on considère donc une solution Cs initialisée à Cs_0 et évaluée avec la fonction critère f dont on veut maximiser la valeur. La recherche opérée ici est une recherche locale.

Nous avons noté au cours des parties précédentes que l'utilisation d'une fonction critère de type "minmin" sur un problème dont le nombre de satellites pouvait varier, engendrait une forte irrégularité du critère. En effet, le changement de valeur d'un seul paramètre peut engendrer une désynchronisation d'un satellite et ainsi faire chuter la pire des performances de manière très importante.

Plaçons nous par exemple dans le cadre de la navigation et supposons que pour une constellation donnée tous les points de la zone d'intérêt aient en visibilité 4 satellites à tout instant. L'évaluation d'une telle constellation à l'aide d'un critère de type minmin donne une valeur, plus ou moins bonne, de la constellation. Une modification pendant le processus d'optimisation d'un satellite sur son orbite peut engendrer une visibilité restreinte à trois satellites, pour un ou plusieurs points, au cours de la simulation. Sous de telles hypothèses la valeur du critère chute considérablement puisque la précision en ces points devient nulle. Cette constatation est identique si on retire un satellite de la constellation.

En appliquant un calcul de moyenne (ou de médiane) plutôt qu'un calcul de pire cas, on tend à réduire l'impact d'une telle modification. Si de plus le nombre de satellites est statique, la régularité du critère devient suffisante pour appliquer un algorithme inspiré de méthodes classiques.

L'objectif de ce niveau étant la convergence vers une solution localement optimale (ou sous optimale) nous allons mettre en oeuvre un algorithme de recherche de plus grande pente.

L'inconvénient rencontré dans sa mise en oeuvre est la présence d'un nombre important d'optimums locaux. Pour tenter de sortir de ces optimums locaux nous avons mis en oeuvre un processus

de redémarrage qui est basé sur un tirage aléatoire des paramètres de position du satellite (Ω et M).

12.3 Mise en oeuvre de la descente de plus grande pente

12.3.1 Définition du voisinage

Pour mettre en oeuvre la descente de plus grande pente, nous avons quelque peu besoin de l'adapter. En effet, la théorie nous indique que la recherche de la pente maximale se fait en étudiant le gradient de la fonction d'évaluation. Or dans notre cas, l'expression analytique de cette fonction n'est pas connue (rien ne nous assure d'ailleurs que le critère est dérivable en un quelconque sens). Nous allons donc converger vers un optimum en cherchant dans le voisinage $V(C_s)$ de la solution courante C_s la solution de plus grande évaluation (nous n'évaluerons donc pas de gradient).

Par ailleurs, nous avons besoin de définir un voisinage de la solution courante. Celui-ci sera défini par discrétisation des plages de variation des paramètres orbitaux. Pour mettre en oeuvre cette discrétisation, des problèmes classiques se posent :

- Le pas de discrétisation doit être suffisant pour que la perturbation appliquée à la solution courante soit suffisante pour qu'une variation soit ressentie au niveau de son évaluation.
- Mais celui-ci ne doit pas être trop important pour que la notion de voisinage garde tout son sens (pas de changement de structure de la constellation par exemple).

A ces problèmes classiques, il faut encore ajouter ceux propres au domaine d'application à savoir une influence différente des paramètres orbitaux sur les orbites. par exemple, les paramètres les plus influents (a , i) devront être discrétisés de manière plus fine. Enfin, il faut noter que le pas de discrétisation ne sera pas identique sur toute la plage de variation d'un même paramètre. Par exemple, une variation de $10km$ sur le paramètre h de valeur $25000km$ a peu d'influence (sur une période d'orbite) alors que la même variation à $400km$ a une influence beaucoup plus grande.

Ces différents problèmes seront résolus en appliquant une mise à l'échelle des paramètres orbitaux en fonction de leur plage de variation (cf hypothèses section 12.1) et de leur nature physique.

Cette discrétisation des 6 paramètres orbitaux définit un voisinage du type de celui présenté figure 12.1 (dans le cas de cette figure, seuls deux paramètres a et b sont représentés pour illustrer le principe). Dans notre cas, pour minimiser les risques d'irrégularité de la fonction d'évaluation, le voisinage de la solution courante ne sera composé que des voisins sur les axes principaux (figure 12.2). Autrement dit un seul paramètre pourra varier après une itération de l'algorithme.

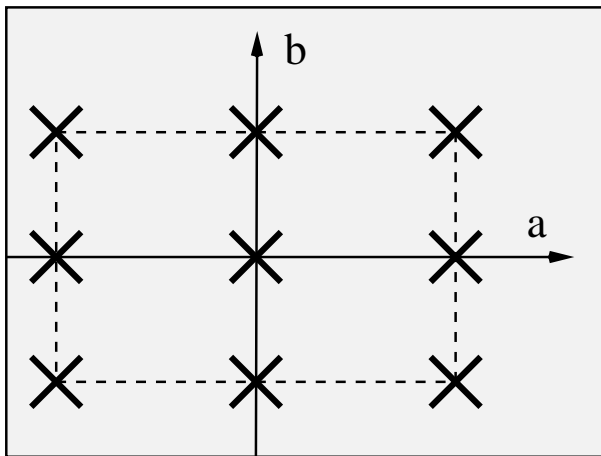


FIG. 12.1 – *Voisinage complet*

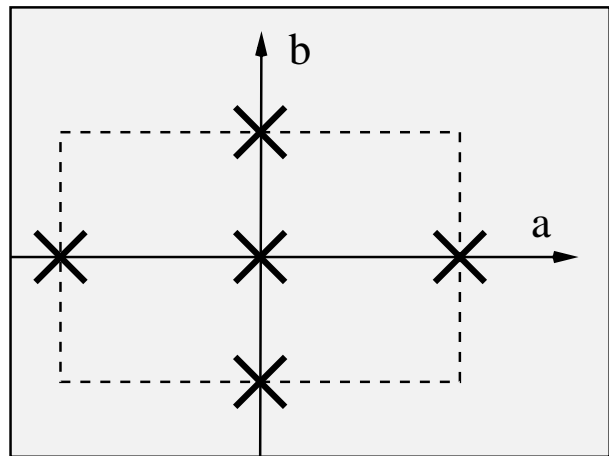


FIG. 12.2 – *Voisinage restreint*

12.3.2 Amplitude de la perturbation

Après avoir déterminé la direction de plus forte pente, il reste à déterminer de manière quantitative la perturbation à appliquer au paramètre. Pour minimiser les évaluations, on définit un pas adapté de variation pour chaque paramètre et on progresse le long de l'axe choisi tant que l'on améliore la solution.

12.3.3 Justification

Au cours de cette phase, nous devons jouer sur le compromis entre le temps passé à évaluer le critère coûteux, et la qualité de la solution trouvée. Dans l'absolu, nous devrions faire une recherche de plus forte pente à chaque modification d'un paramètre or il faudrait dans ce cas évaluer beaucoup plus de constellations. Plusieurs tests ont été réalisés avant d'adopter cette version de la méthode, des résultats sont donnés dans le tableau 12.1.

Ces données correspondent à la résolution du problème suivant :

- Maximiser le pourcentage de couverture d'une zone Δ .
- Constellation considérée comme optimale à 4 satellites coplanaires (valeur 6.3797).
- Solution initiale P_1 : solution proche de l'optimum global (seule la position des satellites change sur l'orbite).
- Solution initiale P_2 : solution proche d'un optimum local (on change en plus l'altitude des satellites).
- Solution initiale P_3 : solution éloignée de l'optimum global (les satellites ne sont plus coplanaires).

Pour ces tests, la solution initiale est choisie proche d'un optimum local et nous comparons la vitesse de convergence (en terme de temps de calcul) à la qualité de la solution trouvée. Les problèmes P_i^1 sont traités par la résolution théorique (2^n voisins) et les problèmes P_i^2 par notre méthode ($2n$ voisins).

Dans le tableau 12.1, P_1^1, P_1^2, P_2^1 convergent vers l'optimum global alors que les autres convergent vers des optimums locaux. Dans la majeure partie des tests effectués, notre méthode converge vers

Problème	Nombre d'itérations	Nombre d'évaluations	valeur solution
P_1^1	32	640	6.3797
P_1^2	61	73	6.3797
P_2^1	35	500	6.3797
P_2^2	75	71	6.2452
P_3^1	38	837	5.1561
P_3^2	87	135	3.3879

TAB. 12.1 – Influence des voisinages

une solution acceptable (même si elle est sous optimale). Dans certains cas la solution trouvée est de mauvaise qualité, mais le temps de convergence est fortement réduit. Cela vient du fait que dans le second cas d'application, l'optimum local de convergence n'est pas le même que celui trouvé avec la première méthode.

Le fait que la solution converge en plus ou moins d'itérations vers un optimum local est important car il en existe un grand nombre et l'objectif (à ce niveau) est de trouver une solution localement acceptable : il faut donc converger vers un maximum d'optimums (cf section suivante) pour en extraire un satisfaisant.

12.4 Rencontre d'optimum locaux

12.4.1 Définition de l'optimum local

De manière générale, un optimum local est défini par un voisinage centré sur celui-ci et à l'intérieur duquel il n'existe pas de solution meilleure.

Soi Cs_0 une constellation,

$$Cs_0 \text{ maximum local} \equiv \exists V(Cs_0) | Cs_0 \in V(Cs_0), \forall Cs \in V(Cs_0), f(Cs_0) \geq f(Cs)$$

Dans notre cas d'application nous allons définir un paramètre de sensibilité S_{SD} qui nous permet de définir un voisinage $V(Cs_0)$ tel que :

$$\forall Cs \in V(Cs_0), (f(Cs_0) + S_{SD}) \geq f(Cs)$$

Cette définition nous permet d'arrêter le processus de descente lorsque la progression de la solution n'est pas suffisante. Dans le cas d'une recherche par descente sur des axes principaux discrétisés, le voisinage de la solution courante Cs_0 selon l'axe du paramètre α ($\alpha \in \{a, e, i, \omega, \Omega, M\}$) est défini par un couple $\{\alpha_1, \alpha_2\}$. Lorsque $(f(Cs_0) + S)$ sera supérieur à l'évaluation de tous les voisins, le processus de descente sera considéré comme terminé.

12.4.2 Principe de redémarrage

Lorsque la descente de plus grande pente rencontre un optimum local, il faut trouver un moyen de sortir de son bassin attractif.

Nous avons décidé de mettre en place un redémarrage en tirant aléatoirement la valeur de certains paramètres. Si les valeurs de a et i obtenue après la phase de recherche de pente a une importance quant aux caractéristiques de l'orbite (chapitre 11), celle des paramètres de position du satellite (Ω, M) peut quant à elle être modifiée sans impact sur la forme de l'orbite.

Nous avons donc décidé d'appliquer des perturbations aléatoires sur ces paramètres (Ω, M) pour sortir du bassin de l'optimum local. Lorsqu'une meilleure solution est trouvée, le processus de descente est remis en application pour affiner la valeur des paramètres de position ainsi que la valeur des paramètres a et i . Ces modifications sur la position du satellite permettent de placer celui-ci par rapport aux autres en testant différentes combinaisons (éventuellement éloignées les unes des autres).

12.5 Durée de la recherche

La recherche est donc une alternance de deux phases : la phase d'intensification au travers de la recherche de plus grande pente et la phase de diversification au travers du redémarrage. Ces phases s'alternent pendant un certain nombre d'itérations N_{SD} que l'on considère fixé à ce niveau de l'algorithme.

La valeur de la sensibilité permettant la détection d'un optimum local est elle aussi fixée ainsi que le nombre de tirages aléatoires réalisés lors du redémarrage. Il faut noter que la sensibilité conditionne les durées d'alternance des deux phases. Nous verrons que les niveaux supérieurs vont jouer sur la durée de ces phases pour guider la recherche.

12.6 Bilan

La méthode appliquée ici ne constitue pas le coeur de notre algorithme. La mise en oeuvre proposée est relativement simple et permet de rechercher une solution sous-optimale. Ce qu'il faut

retenir c'est l'alternance de deux phases à ce niveau de l'algorithme qui permet d'explorer un sous-espace afin que les niveaux supérieurs puissent évaluer sa capacité à résoudre le problème.

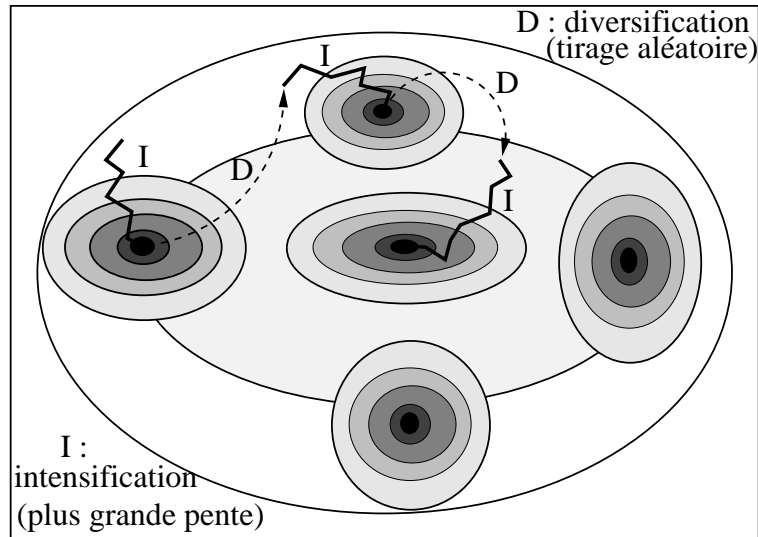


FIG. 12.3 – Illustration de l'algorithme du niveau numérique

Le niveau heuristique permet :

une initialisation de la constellation (Cst) (12.1)

une définition des limites de la zone de recherche (12.1)

tantque le nombre maximal d'itérations n'est pas atteint (12.5)

Évaluer le voisinage de Cst (12.3.1).

Si Cst est un optimum local alors

Appliquer le principe de redémarrage (diversification) (12.4.2)

Sinon

Sélectionner le meilleur voisin Cst^+ (intensification) ($Cst < -Cst^+$)

FinSi

fin tantque

TAB. 12.2 – Algorithme du niveau numérique

Chapitre 13

Résolution : le niveau métaheuristique

13.1 Introduction

Nous sommes ici au niveau le plus haut dans l'algorithme. C'est ici que la recherche va être orchestrée. Selon le formalisme décrit dans le chapitre 11, ce niveau répond à la question : "quelles configurations explorer et comment les explorer?"

Nous allons présenter au cours de ce chapitre la correspondance entre les données et les structures de notre problème et le formalisme de la méthode utilisée à ce niveau : la recherche tabou présentée dans le chapitre 8. Cet algorithme a été retenu pour implémenter ce niveau car c'est une méthode de recherche locale qui permet de nous affranchir de la lourdeur des méthodes faisant évoluer une population de solutions. Elle intègre de plus des aspects avancés tels que la notion de mémoire et les oscillations stratégiques utiles pour guider la recherche tout au long des itérations.

Nous allons successivement nous intéresser au cours de ce chapitre :

- à la définition de l'espace de recherche, en exprimant notamment sa complexité,
- aux transitions utilisées pour accéder au successeur de la solution courante,
- au voisinage d'une configuration,
- aux critères utilisés pour évaluer les éléments de notre problème (satellite, constellation, classe, configuration),
- aux différents aspects de la mémoire,
- aux conditions de mise à l'état tabou d'une configuration,
- au choix du successeur avec notamment la présentation de la liste de candidats et des critères d'aspiration,
- à l'application des transitions,
- à la gestion des oscillations stratégiques avec notamment l'évaluation de la progression d'une solution.

On rappelle les notations adoptées au chapitre précédent :

- On note $ODB = \{Cl_1, \dots, Cl_m\}$, la base de données d'orbites composées des m classes.
- On note $Cf_k = (Ns_1^k, \dots, Ns_m^k)$, la k^{ime} configuration composée de Ns_i^k représentants (satellites) de la classe Cl_i ($i \in [1, m]$).

13.2 L'espace de recherche

13.2.1 Définition

Nous sommes au niveau d'abstraction le plus élevé de l'algorithme, l'espace de recherche n'est donc plus composé de constellations comme pour le niveau bas (niveau numérique) mais de configurations (chapitre 11).

Comme indiqué au chapitre précédent, si l'espace des constellations permet de recouvrir l'ensemble des solutions possibles et ce de manière bijective (à $6n$ paramètres orbitaux correspond une et une seule constellation et à une constellation correspond un et un seul jeu de $6n$ paramètres orbitaux). Ce n'est plus le cas lorsque l'on passe à une représentation sous forme de configurations. A une configuration correspond un ensemble de solutions (constellations), et une constellation peut appartenir à plusieurs configurations. Cette particularité permet l'introduction, via la base de connaissances ODB , d'une expertise en découpant de manière non uniforme l'espace des constellations (figure 13.1).

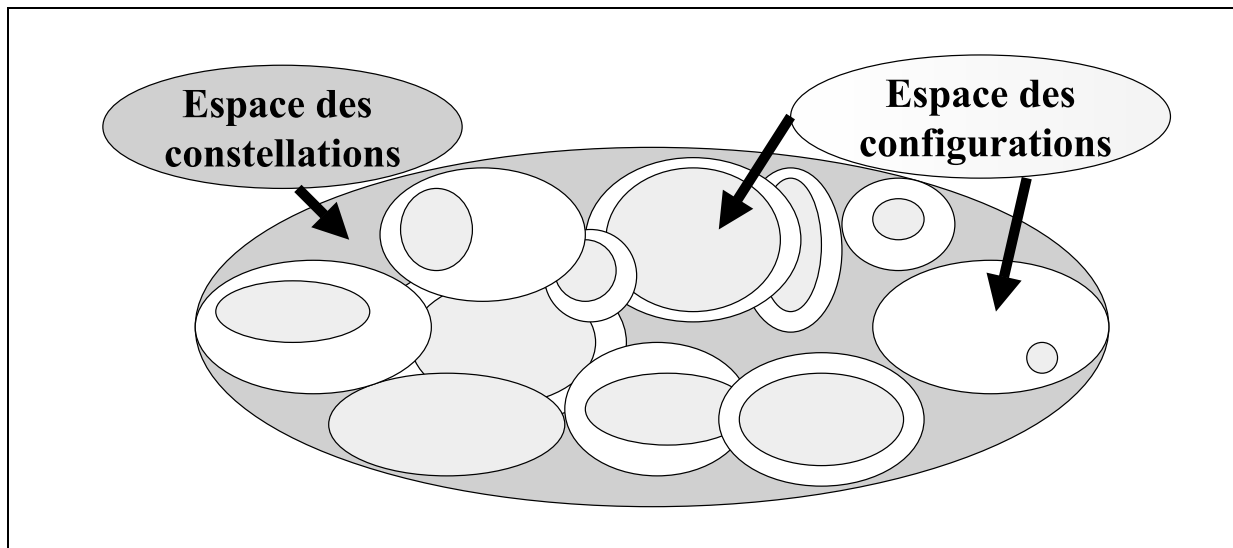


FIG. 13.1 – Découpage de l'espace des constellations

L'algorithme tabou navigue donc parmi un ensemble de configurations définies à partir des classes issues de l'ODB.

De manière générale, on cherche une constellation dont on ignore le nombre n de satellites mais dont on sait donner un minorant n_{min} et un majorant n_{max} . Ces nombres peuvent définir un intervalle de variation important pour n mais plus la valeur $(n_{max} - n_{min})$ sera petite et plus la recherche sera facilitée.

Sous ces hypothèses, l'ensemble des configurations accessibles est :

$$CF = \bigsqcup_{n=n_{min}}^{n_{max}} Cf_n$$

avec

$$Cf_n = \{(Ns_1, \dots, Ns_m) \mid \forall i \in [1, m] Ns_i \geq 0, \sum_{j=1}^m Ns_j = n\}$$

13.2.2 Évaluation de la complexité de l'espace

Si on prend l'exemple d'une base de donnée composée de deux classes ($ODB = \{Cl_1, Cl_2\}$) avec $n \in \{3, 4, 5\}$ alors CF est l'ensemble :

$$\begin{aligned} CF = & \{(3,0), (2,1), (1,2), (0,3)\} \cup \\ & \{(4,0), (3,1), (2,2), (1,3), (0,4)\} \cup \\ & \{(5,0), (4,1), (3,2), (2,3), (1,4), (0,5)\} \end{aligned}$$

Avec seulement 2 classes et trois valeurs possibles pour n , on obtient 15 configurations possibles. La combinatoire augmente de manière exponentielle avec le nombre de classes et le nombre de satellites.

Nous allons maintenant évaluer la cardinalité $P_m^{n_{min}, n_{max}}$ correspondant à l'ensemble des choix possibles de n (n variant de n_{min} à n_{max}) et choisis parmi m classes (une classe peut donc être choisie de 0 à n_{max} fois). On note par ailleurs T_m^n l'ensemble des choix possibles pour n satellites (n fixé) parmi m classes et on a $Card(T_m^n) = m^n$.

Or

$$P_m^{n_{min}, n_{max}} = \bigcup_{n=n_{min}}^{n_{max}} T_m^n$$

et

$$Card(P_m^{n_{min}, n_{max}}) = \sum_{n=n_{min}}^{n_{max}} Card(T_m^n) = \sum_{n=n_{min}}^{n_{max}} m^n$$

On a donc pour $n_{min} = 0$:

$$Card(P_m^{0, n_{max}}) = \sum_{n=0}^{n_{max}} m^n = \frac{m^{n_{max}+1} - 1}{m - 1} \quad (13.1)$$

Or

$$P_m^{n_{min}, n_{max}} = \bigcup_{n=n_{min}}^{n_{max}} T_m^n = \bigcup_{n=0}^{n_{max}} T_m^n \setminus \bigcup_{n=0}^{n_{min}-1} T_m^n = P_m^{0, n_{max}} \setminus P_m^{0, n_{min}-1}$$

d'où

$$Card(P_m^{n_{min}, n_{max}}) = Card(P_m^{0, n_{max}}) - Card(P_m^{0, n_{min}-1})$$

c'est à dire, d'après l'équation 13.1,

$$Card(P_m^{n_{min}, n_{max}}) = \frac{m^{n_{max}+1} - 1}{m - 1} - \frac{m^{n_{min}} - 1}{m - 1} = \frac{m^{n_{max}+1} - m^{n_{min}}}{m - 1} \quad (13.2)$$

Dans le cas d'une constellation comportant entre 15 et 20 satellites, ce qui est une constellation de petite taille si on souhaite par exemple couvrir la terre pour des besoins de navigation, et si on considère une base de donnée composée de 6 classes, le nombre de combinaisons est 4387296091078656. L'espace de recherche est donc immense et fortement combinatoire et il est inconcevable de l'explorer de manière exhaustive. L'objectif de ce niveau de l'algorithme est de parcourir le moins possible de configurations avant de trouver la configuration optimale.

13.3 Les transitions

Après avoir défini la composition de l'espace de recherche et son étendue, nous allons définir la manière dont on passe d'une solution à son successeur. Dans notre cas, il y a trois types de transitions permettant de passer de la configuration courante Cf^c à son successeur Cf^+ (figure 13.2):

- T_0 : on ajoute un représentant (un satellite) d'une classe à la configuration courante.
- T_1 : on retire un représentant (un satellite) d'une classe à la configuration courante.
- T_2 : on change la classe d'appartenance d'un représentant (un satellite) d'une classe de la configuration courante.

Ces trois types de transitions sont la base d'un ensemble plus important de transitions. En effet, la transition T_0 , correspondant à l'ajout d'un satellite, admet autant de dérivées qu'il existe de classes dans la base ODB (T_0 se dérive en $\{T_0^1, \dots, T_0^m\}$). Dans le cas des types de transitions T_1 et T_2 , il faut noter qu'il n'y a pas de distinction entre les différents représentants d'une même classe (la distinction étant faite ultérieurement). Si les satellites Sat_1 et Sat_2 de la constellation Cs sont deux représentants de la classe Cl_k , alors supprimer Sat_1 ou Sat_2 correspond à la même transition T_1^k . Soit une configuration $Cf = \{Ns_1, \dots, Ns_m\}$ telle que $n_{min} < \sum_{i=1}^m Ns_i < n_{max}$. L'ensemble des transitions dérivées est :

- $\{T_0^i\}, i \in [1, m]$, T_0^i correspond à l'ajout d'un représentant de la classe Cl_i .
- $\{T_1^i\}, i \in \{i_1, \dots, i_p\}$, T_1^i correspond au retrait d'un représentant de la classe Cl_i .

- $\{T_2^{i,j}\}, i \in [1,m], j \in \{i_1, \dots, i_p\}, T_2^{i,j}$ correspond au remplacement d'un représentant de la classe Cl_i par un représentant de la classe Cl_j .

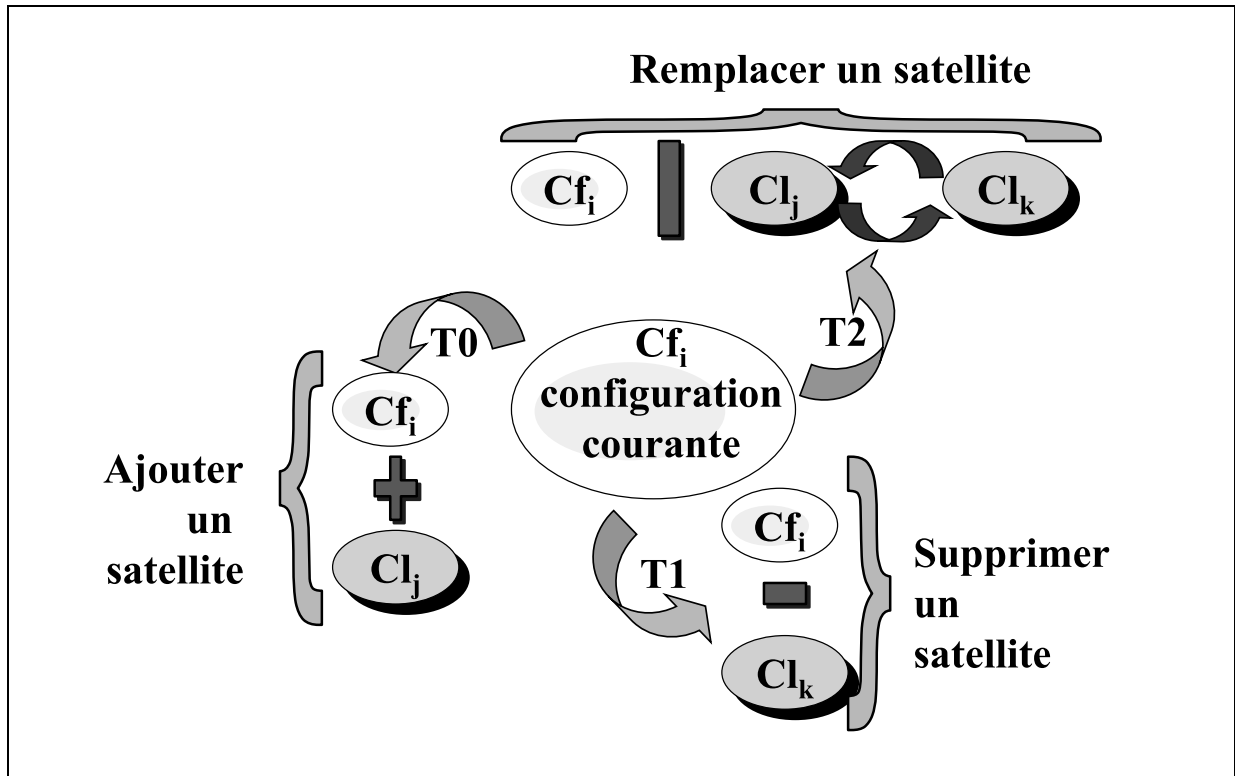


FIG. 13.2 – Les transitions

13.4 Le voisinage

13.4.1 Définition

Nous allons présenter ici le voisinage (pris comme un ensemble de configurations) engendré par les transitions décrites précédemment. Le voisinage simple (sans notion d'état tabou ou de liste de candidat) est l'ensemble $V(Cf^c)$ des configurations accessibles par application d'une transition. Si on considère que $Cf^c = (Ns_1, \dots, Ns_m)$, alors le voisinage est l'ensemble des configurations $Cf' = (Ns'_1, \dots, Ns'_m)$ telles que

- $\exists i \in [1,m], Ns'_i = Ns_i - 1$ et $\forall j \in [1,m], j \neq i, Ns'_j = Ns_j$
- $\exists i \in [1,m], Ns'_i = Ns_i + 1$ et $\forall j \in [1,m], j \neq i, Ns'_j = Ns_j$
- $\exists i \in [1,m], Ns'_i = Ns_i - 1$ et $\exists j \in [1,m], j \neq i, Ns'_j = Ns_j + 1$ et $\forall k \in [1,m], k \neq i, k \neq j, Ns'_k = Ns_k$

Par ailleurs, $\forall Cf' \text{ in } V(Cf^c), n_{min} \leq \sum_{i=1}^m Ns_i' \leq n_{max}$.

13.4.2 Taille du voisinage

Cas général

Afin de mieux percevoir la complexité du problème, nous allons évaluer la taille du voisinage d'une configuration quelconque. Elle est calculée en intégrant la contribution de chacune des transitions définies précédemment.

Avec les notations précédentes ($P_m^{n_{min}, n_{max}}$ et $Cf = (Ns_1, \dots, Ns_m)$), on définit N comme étant le nombre de classes ayant au moins un représentant dans Cf ($Ns_i > 0$) et $n = \sum_{i=1}^m Ns_i$ avec $n_{min} < n < n_{max}$.

Le nombre de voisins de Cf est donc $m(N + 1)$:

- m voisins définis par T_0 (on peut ajouter un représentant de n'importe quelle classe).
- N voisins définis par T_1 (on peut retirer un représentant de n'importe quelle classe utilisée).
- $(m - 1) * N$ voisins définis par T_2 (pour les N classes utilisées on peut choisir parmi les $m - 1$ classes restantes).

Cas des bordures de l'espace de recherche

Le nombre n de représentants étant borné par la donnée de n_{min} et n_{max} le voisinage d'une configuration telle que $\sum_{i=1}^m Ns_i = n_{min}$ ou $\sum_{i=1}^m Ns_i = n_{max}$ est donc modifié par rapport au cas général précédent. Un certain nombre de configurations sort en fait du voisinage et ce dernier se trouve être de complexité inférieure. On a les cas particuliers suivants :

- si $\sum_{i=1}^m Ns_i = n_{max}$, le nombre de voisins est $N + N * (m - 1) = Nm$ (contribution de T_1 et T_2).
- si $\sum_{i=1}^m Ns_i = n_{min}$, le nombre de voisins est $m + (m - 1) * N$ (contribution de T_0 et T_2).

13.5 De multiples critères

Afin de choisir à chaque itération la transition à appliquer, nous avons besoin d'évaluer chacun des éléments définis dans notre système : les satellites, les classes et les configurations (ces évaluations tiennent implicitement compte de l'évaluation de la constellation). Par ailleurs, chacune des valeurs décrites ici donneront lieu à un attribut associé à l'élément évalué. Par abus de langage nous désignerons l'attribut par la valeur associée.

13.5.1 Évaluation d'une constellation

Nous rappelons ici que la valeur d'une constellation est directement liée au critère d'optimisation défini par l'utilisateur. C'est un critère coûteux, basé sur le processus de simulation décrit dans le chapitre 3.

Cette valeur est la plus fiable des évaluations définies dans cette section. Elle rend compte des performances réelles de la constellation et de son aptitude à résoudre le problème. Les critères utilisés sont décrits dans les chapitres 4 et 5.

L'utilisation de cette évaluation est faite au niveau le plus bas pour optimiser les jeux de paramètres orbitaux (chapitre 12). On note f la fonction d'évaluation d'une constellation.

13.5.2 Évaluation d'un satellite

Un des inconvénients du critère précédent, déjà exprimé au cours des chapitres précédents, est le manque d'informations retournées sur les satellites composant la constellation. On sait en effet si la constellation répond au besoin mais sans guère plus d'informations quantitatives ou qualitatives. Le biais selon lequel il faut modifier la constellation est donc difficile à estimer.

Pour pallier ce manque, nous avons décidé d'évaluer chacun des satellites de la constellation pendant la simulation afin de rendre compte de leur contribution. Cette évaluation est multiple pour permettre un choix plus fin sur les satellites.

Pour les problèmes considérés (télécommunications et navigation), la notion de couverture est très importante et l'évaluation des satellites est essentiellement basée sur elle. Nous allons calculer 4 valeurs ($V_{s1}, V_{s2}, V_{s3}, V_{s4}$) au cours de la simulation :

- V_{s1} : le nombre de points d'intérêt (spatio-temporel) couverts au cours de la simulation, où le satellite considéré était critique. Dans le cas d'une couverture simple, un satellite est dit critique s'il est seul à couvrir le point d'intérêt à l'instant considéré. Dans le cas de la navigation, le satellite est considéré comme critique s'il y a moins de 4 satellites en visibilité de ce point au même moment.
- V_{s2} : le nombre de points d'intérêts (spatio-temporel) couverts au cours de la simulation où le satellite considéré n'était pas critique.
- $V_{s3}(V_{s4})$: le nombre de fois où le satellite participe à des calculs de performance locale dont la valeur est supérieure (inférieure) au seuil fixée. Ces dernières valeurs ne sont utilisées que pour des problèmes de navigation afin de savoir si le satellite est utilisé de manière performante ou pas. Dans le cas des télécommunications, la bonne performance correspond à couvrir le point considéré donc ($V_{s3} = V_{s1} + V_{s2}$) et ($V_{s4} = 0$).

Au niveau bas de l'algorithme, les valeurs calculées peuvent être assimilées à la notion de mémoire fréquentielle de la recherche tabou.

On calculera, à l'aide de ces valeurs, et d'une fonction d'agrégation (somme pondérée dépendant de l'application traitée), une valeur V_{st} utilisée lors de l'application des transitions pour sélectionner les satellites. Lorsque nous parlerons de valeur d'un satellite dans le reste du document, nous ferons allusion à cette dernière valeur.

13.5.3 Évaluation à posteriori d'une configuration

Après avoir parcouru une configuration, on attribue à celle-ci une valeur qui rend compte de son intérêt. Cette valeur (V_{cf}) est la valeur moyenne des constellations (Cs) visitées pendant la recherche de bas niveau :

$$V_{cf}^1(Cf) = \frac{1}{Card(\Pi_{Cf})} \sum_{Cs \in \Pi_{Cf}} (f(Cs)).$$

Π_{Cf} étant l'ensemble des constellations visitées au sein de la configuration Cf .

13.5.4 Évaluation d'une classe

La forte combinatoire de l'espace de recherche est en partie engendrée par le nombre de classes de l'*ODB*. Il faut donc pouvoir choisir parmi ces classes, celles qui présentent un intérêt pour le problème posé et ainsi guider la recherche. Nous allons donc évaluer chacune d'entre elles au cours de la recherche de la manière suivante : la valeur d'une classe (Cl_i) est la valeur moyenne des configurations visitées, pondérée par le nombre de représentants de la classe dans la configuration considérée (une classe ne peut avoir de valeur représentative que si elle a été utilisée au moins une fois).

$$V_{cl}(Cl_i) = \frac{1}{\sum_{j=1}^N Ns_i^j} * \sum_{j=1}^N Ns_i^j * V_{cf}^1(Cf_j)$$

Avec $\Delta = \{Cf_i, i \in [1, N]\}$ l'ensemble des configurations visitées.

13.5.5 Évaluation à priori d'une configuration

Lorsque le processus de recherche tabou doit passer d'une configuration à une autre, il doit évaluer le voisinage de la configuration courante. La difficulté d'une telle entreprise est le temps d'évaluation d'une configuration. Une configuration définissant une région de l'espace des constellations, il faudrait évaluer de la manière décrite dans la section 13.5.3 (et donc appliquer l'algorithme 12.2 du chapitre 12) toutes les configurations voisines avant de choisir le successeur.

On introduit donc une évaluation dite à *priori* dont le calcul sera moins coûteux que le précédent, ce qui permettra d'évaluer chacune des configurations du voisinage.

On définit pour cela une distance maximale d_{max} entre la configuration à évaluer (Cf) et les configurations intégrées dans le calcul. Soit $\Phi = \{Cf' | Cf' \in \Delta, d(Cf, Cf') \leq d_{max}\}$. La valeur à *priori*

de la configuration Cf est :

$$V_{cf}^2(Cf) = \frac{1}{Card(\Phi)} \sum_{Cf' \in \Phi} \left(V_{cf}^1(Cf') \cdot \frac{M(Cf)}{M(Cf')} \right)$$

$$M(Cf) = \frac{1}{\sum_{i=1}^m N_{s_i}} \sum_{k=1}^m (N_{s_k} \cdot V_{cl}(Cl_k))$$

$M(Cf)$ est la valeur moyenne des classes présentes dans la configuration Cf , pondérée par le nombre de représentants de la classe. Le coefficient $\frac{M(Cf)}{M(Cf')}$ correspond à l'augmentation (ou la diminution) à laquelle on doit s'attendre en passant de la configuration Cf' à la configuration Cf .

L'évaluation *à posteriori* présentée précédemment est plus fiable que l'évaluation *à priori*.

13.5.6 Bilan

Le tableau 13.1 récapitule les évaluations présentées au cours de cette section :

élément	évaluation
satellite	$V_{s1}, V_{s2}, V_{s3}, V_{s4}$ (simulation), V_{St} (agrégation)
classe d'orbite	V_{cl} (contribution dans les configurations visitées)
constellation	f (simulation)
configuration	V_{Cf}^1 (<i>à posteriori</i>), V_{Cf}^2 (<i>à priori</i>)

TAB. 13.1 – *Bilan des évaluations*

13.6 La mémoire

Nous allons détailler ici les différents aspects de la mémoire en précisant les structures que nous utilisons pour notre algorithme.

13.6.1 La récence

Cette partie de la mémoire à court terme permet de ne pas revenir sur des choix réalisés récemment. Dans notre algorithme, il s'agit de ne pas revisiter des configurations récemment visitées et de ne pas supprimer un satellite récemment ajouté ou modifié.

Deux listes d'éléments sont nécessaires pour éviter de revenir sur ces choix. Il faut en effet mémoriser la liste LT_{Cf}^1 des k dernières configurations visitées ainsi que la liste LT_{St} des k' derniers représentants (satellites) de classes ajoutés.

Pour s'en convaincre nous allons considérer le cas simple suivant :

$$- ODB = \{Cl_1, Cl_2\}$$

- à une itération i la configuration Cf_i est composée d'un représentant de la classe Cl_1 , noté Sat_1^1 , et d'un représentant de la classe Cl_2 , noté Sat_2^1 . On a donc $Cf_i = \{1,1\}$.
- La transition appliquée pour passer à la configuration suivante consiste à ajouter un représentant Sat_1^2 de la classe Cl_1 . La nouvelle configuration est donc $Cf_{i+1} = \{2,1\}$.

Si on ne mémorise que les satellites ajoutés, la suppression du satellite Sat_1^2 n'est pas possible mais en revanche, la suppression de Sat_1^1 l'est. Il est donc possible de visiter à nouveau la configuration $\{1,1\}$.

Si on ne mémorise que les configurations et que l'on passe de la configuration Cf_{i+1} à Cf_{i+2} en ajoutant un second représentant de la classe Cl_2 ($Cf_{i+2} = \{2,2\}$), on peut alors supprimer le satellite Sat_1^2 récemment ajouté puisque la configuration $\{1,2\}$ n'est pas présente dans la liste des configurations visitées.

On définit donc deux listes servant à mémoriser la récence des éléments ajoutés ou visités. Nous verrons que lors des phases de diversification, l'algorithme a besoin de connaître les classes qui ont été récemment utilisées. Ce besoin ne nécessite pas l'introduction d'une autre liste dans la mesure où la liste LT_{cf}^1 permet de générer la liste des classes les plus récemment utilisées.

13.6.2 La qualité

L'aspect qualité de la mémoire se répercute à travers l'évaluation des satellites et des classes. Au niveau de la classe, la qualité est représentée par l'attribut stockant la valeur V_{cl} . Au niveau du satellite, on utilise les quatre valeurs $(V_{s1}, V_{s2}, V_{s3}, V_{s4})$ pour affecter une valeur à un attribut (calculer par une fonction d'agrégation dépendant du problème traité). De manière générale et pour des problèmes de télécommunication, on pose $V_{st} = \omega_1.V_{s1} + \omega_2.V_{s2}$, ω_1 et ω_2 étant deux poids dont la valeur dépend du problème.

Voici quelques exemples de fonctions d'agrégation :

- Si on souhaite avoir une constellation minimale (en nombre de satellite), pour un problème de télécommunication (monocouverture), on utilisera les poids : $\omega_1 = 0.95$ et $\omega_2 = 0.05$. Ceci aura pour effet de favoriser les constellations dont la majeure partie des satellites est utilisée de manière critique.
- Si on souhaite en revanche avoir une constellation plus robuste (toujours pour un problème de mono-couverture), on préférera les poids suivants : $\omega_1 = 0.8$ et $\omega_2 = 0.2$.

Pour des problèmes de navigation, il faut intégrer les données de V_{s3} et V_{s4} . On peut donc postuler la fonction d'agrégation $V_{st} = \omega_1.V_{s1} + \omega_2.V_{s2} + \omega_3.V_{s3} + \omega_4.V_{s4}$, sachant que $\omega_3\omega_4 = 0$. La notion de robustesse est intégrée de la même manière que pour les télécommunications en jouant sur le rapport $\frac{\omega_1}{\omega_2}$.

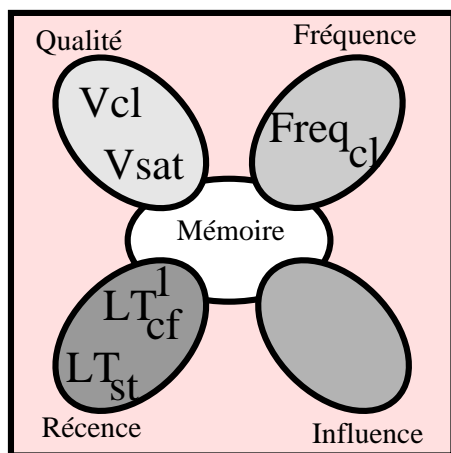
13.6.3 La fréquence

La notion de fréquence est mémorisée au niveau des classes d'orbites. On mémorise en effet la fréquence d'utilisation des classes afin d'orienter la recherche lors de la phase de diversification. Cette aspect de la mémoire implique l'ajout d'un attribut $Freq$ aux classes. Cet attribut, couplé à la qualité (V_{cl}) de la classe, permet d'identifier les "bonnes" et les "mauvaises" classes.

13.6.4 Autres structures

En plus des structures décrites, on mémorise toutes les configurations visitées pendant la recherche (Δ). C'est à dire le m-uplet (Ns_1, \dots, Ns_m) , l'attribut V_{cf}^1 et les paramètres orbitaux de la meilleure constellation visitée au sein de la configuration ($6n$ paramètres).

13.6.5 Bilan



La figure 13.3 synthétise les divers aspects de la mémoire avec leurs applications dans notre algorithme.

On y retrouve les notions de qualité, de récence, de fréquence et d'influence.

On voit par ailleurs, à travers cette figure que nous n'implémentons pas l'aspect influence de la mémoire. Cette notion est représentée par celle de qualité, qui est un cas particulier d'influence ([46]).

FIG. 13.3 – *La mémoire*

13.7 Mise à l'état tabou des configurations

Les différentes structures de mémoire décrites précédemment permettent d'analyser le parcours de la recherche et ainsi guider celle-ci à chaque itération. Pour cela, on restreint le voisinage de la solution courante en mettant un certain nombre de solutions voisines (des configurations à ce niveau de l'algorithme) à l'état tabou (chapitre 8). La mise à l'état tabou d'une configuration peut être déclenchée par plusieurs facteurs que nous allons détailler.

On cherche, dans cette section, à déterminer l'ensemble des configurations tabous dans le voisinage de la configuration $Cf = \{Ns_1, \dots, Ns_m\}$ avec $\sum_{i=1}^m Ns_i = n$. Le voisinage est défini comme dans la section 13.4 (avec toutes les restrictions liées aux bordures de l'espace de recherche).

13.7.1 Utilisation de LT_{cf}^1

Pour éviter de revenir sur des choix récents, on utilise la mémoire à court terme. Plus particulièrement la liste des k dernières configurations visitées (LT_{cf}^1) est mise à l'état tabou.

Par ailleurs, lorsque l'on est en phase de diversification, on cherche à écarter les configurations de même nature que les dernières visitées. Pour cela, on déduit des configurations stockées dans LT_{cf}^1 la liste des classes récemment utilisées. On met ensuite à l'état tabou l'ensemble des configurations voisines de la configuration courante et obtenues en utilisant une de ces classes. Ces configurations sont déduites de la configuration courante par application d'une transition de type T_0 ou T_2 . Par exemple, si la classe Cl_i se trouve être une classe récemment utilisée, la configuration obtenue par application de la transition T_0^i est écartée de même que les transitions $T_2^{j,i}$.

13.7.2 Utilisation de l'attribut $Freq$ des classes

On intègre ici l'aspect fréquentiel de la mémoire en utilisant l'attribut $Freq$ des classes d'orbites. Cet attribut mémorise la fréquence d'utilisation de la classe et permet, toujours dans le cadre d'une diversification, d'écarter les classes les plus fréquemment utilisées.

13.7.3 Intégration de l'aspect qualité de la mémoire

L'aspect qualité de la mémoire permet de mettre à l'état tabou des configurations que l'on ne souhaite pas visiter car elles ne sont pas dignes d'intérêt. Deux attributs interviennent ici, l'attribut valeur des classes d'orbite (V_{cl}) et l'attribut valeur à priori des configurations (V_{cf}^2).

On écarte du voisinage les configurations Cf' déduites de la configuration courante en utilisant une classe Cl de faible valeur. On définit pour cela un seuil S_{cl} et on interdit l'utilisation des classes telles que $V_{cl} < S_{cl}$.

De la même manière, le calcul de la valeur à priori permet d'écarter directement un ensemble de configurations. On définit pour cela le seuil S_{cf} et on écarte l'ensemble des configurations telles que $V_{cf}^2 < S_{cf}$.

L'ensemble des configurations mises à l'état tabou à cause d'un facteur qualité est stocké dans la liste LT_{cf}^2 .

13.7.4 Notion de borne tabou

A ce niveau de spécification des facteurs influant sur l'état tabou des configurations, il faut faire une distinction entre les configurations de la liste LT_{cf}^1 et les configurations de la liste LT_{cf}^2 .

On définit pour cela la notion de *borne tabou* qui permet de délimiter un sous-ensemble de configurations que l'on met à l'état tabou.

Une configuration Cf est une *borne supérieure tabou* lorsque l'on a déterminé que le problème ne peut pas être résolu avec la configuration Cf . Dans ce cas, toute configuration Cf' inférieure ($Cf' \preceq Cf$) ne peut satisfaire le problème et est positionnée à l'état tabou.

De la même manière une configuration est une *borne inférieure tabou* si on a l'assurance que le problème peut être résolu à l'aide de la configuration Cf . Dans ce cas, toute configuration Cf' supérieure ($Cf \preceq Cf'$) est considérée comme tabou.

Cette dernière borne assure que l'on va chercher une constellation minimale (en nombre de satellites), puisque l'on ne cherche pas à satisfaire le problème avec une configuration comportant plus de satellites. Si la robustesse des constellations est un critère important de la recherche, cette borne ne doit pas être prise en compte.

Dans notre cas, les configurations présentes dans la liste LT_{cf}^1 ne sont pas considérées comme des bornes tabous. Seule la configuration présente dans la liste a été visitée et donc seule celle-ci est mise à l'état tabou. Par contre, les configurations présentes dans la liste LT_{cf}^2 définissent chacune une borne supérieure tabou. C'est-à-dire que toute configuration inférieure (au sens de la définition donnée dans le chapitre 11), est considérée comme tabou.

Cette utilisation des configurations en tant que borne tabou sera complétée dans la section consacrée aux oscillations stratégiques.

13.8 Choix du successeur

13.8.1 Principe

Après avoir évalué (V_{cf}^2) les configurations non tabous du voisinage de la configuration courante, on réalise une sélection du successeur.

On détermine la prochaine configuration à explorer par un tirage probabiliste. Nous avons décidé d'adopter une recherche tabou probabiliste car la sélection se fait sur un critère seulement "approximativement" fiable.

La méthode peut être affinée en réalisant un tirage uniforme parmi les p meilleures solutions du voisinage permettant ainsi d'abstraire le classement incertain des solutions ([46]).

13.8.2 La liste de candidats

Dans certains cas, lorsque la combinatoire du voisinage est trop grande (section 13.4.2), on établit une liste de candidats au lieu de considérer l'ensemble du voisinage.

On sélectionne les meilleurs éléments du voisinage pour faire partie de la liste de candidats.

13.8.3 Les critères d'aspiration

Nous avons détaillé précédemment la manière classique de constituer la liste de candidats mais on peut ajouter à cette liste un certain nombre de configurations auxquelles on peut donner une grande priorité.

C'est par exemple le cas lorsqu'un satellite Sat_i a une évaluation très faible (voire nulle), on peut ajouter à la liste de candidats la configuration obtenue par suppression d'un satellite de la classe Cl_k (si Cl_k est la classe d'appartenance du satellite Sat_i) même si le retrait du satellite est tabou (Sat_i récemment ajouté par exemple). Si la transition correspondante est sélectionnée, le satellite de plus faible évaluation (Sat_i) sera supprimé (cf section 13.9.2).

De la même manière, lorsqu'un satellite se retrouve sur la bordure d'une classe (chapitre 11), et lorsque l'optimisation de bas niveau tend à éloigner celui-ci du centre de la classe, on cherche à changer sa classe d'appartenance de manière à continuer l'optimisation de son placement de la manière la plus "continue" possible (section 13.9.3).

On cherche donc la classe d'orbite la plus proche (au sens de la distance définie dans la section 11.10.2) de la classe actuelle (et du jeu de paramètres orbitaux du satellite actuel). On choisit donc d'appliquer la transition $T_2^{i,j}$ sans tenir compte de considérations de haut niveau telles que l'évaluation des classes d'orbites. Si une telle tendance est détectée à la fin de la phase d'optimisation de bas niveau, la priorité est donnée à la transition associée.

13.9 Application des transitions

Après avoir sélectionné le successeur de la configuration courante, il reste un certain nombre de paramètres à fixer en fonction de la transition utilisée. De manière générale, lorsque l'on applique une transition pour passer d'une configuration à une autre, on ne modifie que les paramètres orbitaux du représentant (satellite) mis en jeu dans la transition. Les paramètres orbitaux des autres représentants sont donc reconduits d'une configuration à l'autre et seront éventuellement modifiés lors de la recherche de niveau numérique. Par ce procédé, on affine les paramètres orbitaux des satellites au fur et à mesure des visites de configurations.

Mais avant d'appliquer la transition, deux cas peuvent se produire : la configuration candidate est une nouvelle configuration ou alors la configuration candidate est une configuration déjà visitée.

Dans le premier cas, on applique la transition de la manière décrite dans les sections 13.9.1, 13.9.2, 13.9.3. Dans le deuxième cas, on applique le principe décrit dans la section 13.9.4.

13.9.1 Application d'une transition de type T_0

Lorsque l'on ajoute un représentant d'une classe (un satellite), il faut fixer la valeur initiale de ses paramètres orbitaux. La valeur des paramètres (a, i) est fixée en prenant le milieu des intervalles $[a_0, a_1]$ et $[i_0, i_1]$. La valeur des autres paramètres $(\alpha \in \{e, \omega, \Omega, M\})$ est choisie aléatoirement dans l'intervalle $[\alpha_0, \alpha_1]$.

13.9.2 Application d'une transition de type T_1

Lorsque l'on doit sélectionner le satellite à supprimer parmi les représentants de la classe sélectionnée, on utilise la mémoire à court terme et plus particulièrement la liste des satellites récemment ajoutés (LT_{st}) (section 13.6.1). On sélectionne le satellite n'appartenant pas à la liste LT_{st} et de plus petite valeur (valeur V_{st} calculée par une fonction d'agrégation sur $V_{s1}, V_{s2}, V_{s3}, V_{s4}$, section 13.5.2).

13.9.3 Application d'une transition de type T_2

Lorsque l'on change la classe d'appartenance d'un satellite, on est amené à modifier la valeur de ses paramètres orbitaux. Nous allons opérer ce passage d'une classe à une autre en minimisant la variation des paramètres.

Considérons l'application de la transition $T_2^{1,2}$ (la classe d'appartenance passe donc de Cl_1 à Cl_2), la modification d'un paramètre α ($\alpha \in \{a, e, i, \omega, \Omega, M\}$) est opérée de la manière suivante :

- Si $\alpha \in ([\alpha_{min}^1, \alpha_{max}^1] \cap [\alpha_{min}^2, \alpha_{max}^2])$ alors la valeur du paramètre reste inchangée,
- Sinon si $\alpha_{min}^1 > \alpha_{max}^2$ alors le paramètre α prend la valeur α_{max}^2 ,
- Sinon ($\alpha_{max}^1 < \alpha_{min}^2$) le paramètre α prend la valeur α_{min}^2 ,

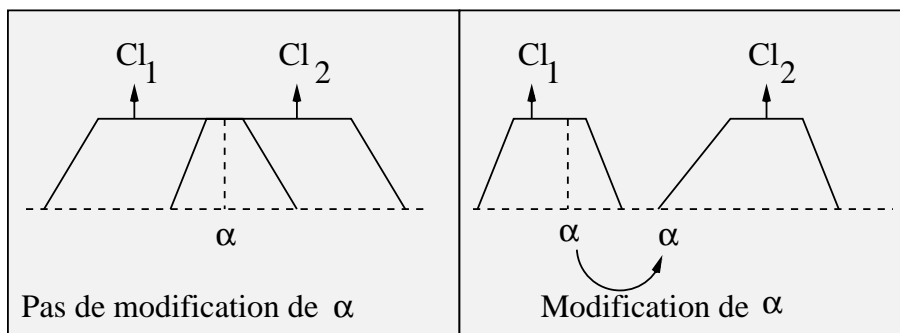


FIG. 13.4 – Application de T_2

13.9.4 Retour vers une configuration déjà visitée

Au cours de la recherche, il est possible de retourner vers des configurations déjà visitées. La non exhaustivité de la recherche de bas niveau, rend ce retour intéressant dans la mesure où il permet d'aller plus loin dans l'exploration des configurations (en explorant plus de constellations).

A ce niveau, même si la configuration est identique, le parcours suivi depuis la dernière visite de celle-ci a engendré des modifications, dans la valeur des paramètres orbitaux des représentants des classes. Deux cas se présentent alors :

- si on est en phase de diversification, on souhaite changer le plus possible la valeur des paramètres orbitaux de la constellation actuelle. Dans ce cas, on réinitialise la constellation avec la meilleure constellation rencontrée lors des explorations précédentes de la configuration (c'est une sorte de redémarrage).

Dans l'exemple de la figure 13.5, le premier parcours part de Cs_0 pour arriver à Cs_2 et la meilleure constellation rencontrée est Cs_1 . Après un retour sur la configuration, le parcours commence en $Cs_3 = Cs_1$ et s'arrête en Cs_4 .

- si on est en phase d'intensification, on souhaite garder le plus possible les paramètres orbitaux de la constellation actuelle. Dans ce cas, on ne change pas les paramètres actuels de la constellation et on applique seulement la transition utilisée pour atteindre la configuration de la manière décrite dans les sections 13.9.1, 13.9.2 et 13.9.3.

Dans l'exemple de la figure 13.6, le premier parcours part de Cs_0 pour arriver à Cs_2 et la meilleur constellation rencontrée est Cs_1 . Après un retour sur la configuration, le parcours commence en $Cs_3 \neq Cs_1$ et s'arrête en Cs_4 .

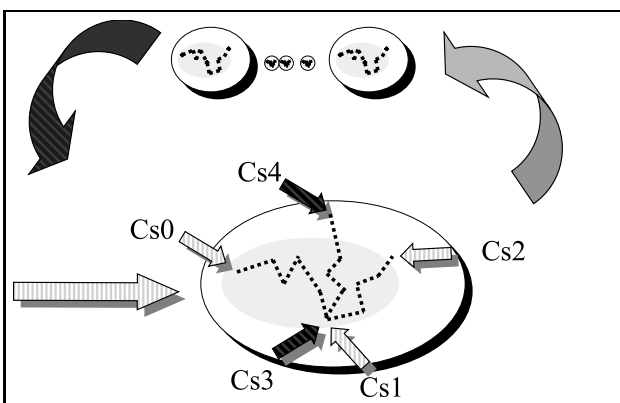


FIG. 13.5 – *Retour avec réinitialisation*

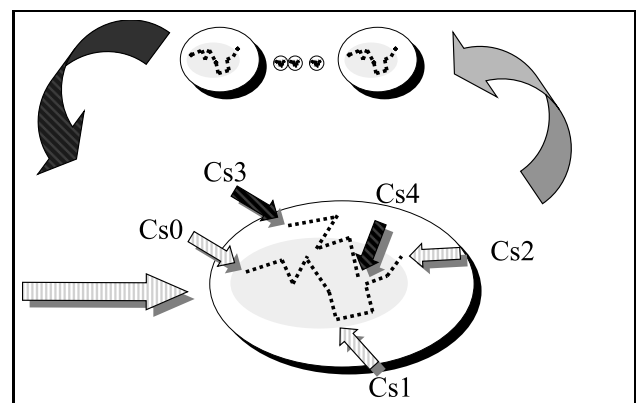


FIG. 13.6 – *Retour sans réinitialisation*

13.10 Les Oscillations stratégiques

13.10.1 Principe

Les oscillations stratégiques permettent de maintenir un objectif à la recherche (intensification) pendant un certain nombre d'itérations (N_{os}). Dans notre cas cet objectif est triple, il s'agit de :

- chercher une configuration ayant un nombre de satellites oscillant autour d'une valeur moyenne fixée, comprise entre n_{min} et n_{max} ,
- d'imposer un certain niveau d'exploration des configurations,
- d'imposer un certain niveau d'exploitation des configurations visitées (S_{SD}, N_{SD}).

Ces trois objectifs sont gérés de manière dynamique durant les N_{os} itérations. Le principe d'une phase des oscillations stratégiques (N_{os} itérations) est de partir d'un champ relativement vaste d'exploration (beaucoup de classes utilisées, large plage de variation du nombre de représentants, recherche de bas niveau limitée) pour ensuite concentrer la recherche vers des zones d'intérêt.

Pour réduire la recherche, il faut écarter des classes d'orbites. Pour cela on augmente la valeur de S_{cl} , on interdit ainsi l'utilisation d'un plus grand nombre de classes. On réduit également la plage de variation du nombre de représentants : n_{min} augmente et n_{max} diminue pour converger vers un nombre moyen de représentants dont la valeur dépend du contexte (histoire, ...). Parallèlement on augmente la durée de la recherche de bas niveau afin d'augmenter l'exploitation des configurations visitées : on augmente donc N_{SD} et on diminue la valeur du seuil S_{SD} . Cette diminution de la valeur du seuil S_{SD} permet d'être moins sensible à la détection des maximums locaux pour conduire à des valeurs de paramètres orbitaux plus précises.

Après les N_{os} itérations correspondant à un effort global d'intensification de la recherche sur une certaine zone de l'espace des configurations, on évalue la **progression** de la solution (section 13.10.2) depuis le début de la phase d'intensification. Si la solution a progressé, on continue la phase d'intensification, sinon, une phase de diversification est entamée afin de reconduire le même type de recherche dans une autre zone. Dans le cas de la diversification, on augmente la plage de variation du nombre de représentants, on intègre plus de classes, et on réduit à nouveau la phase de recherche de bas niveau.

Il faut également noter que la mémoire à long terme est exploitée pour mémoriser la qualité des solutions obtenues au terme de chaque phase et ainsi pouvoir estimer plus finement le nombre moyen de satellites à considérer.

Ces différents phénomènes sont illustrés dans la figure 13.7.

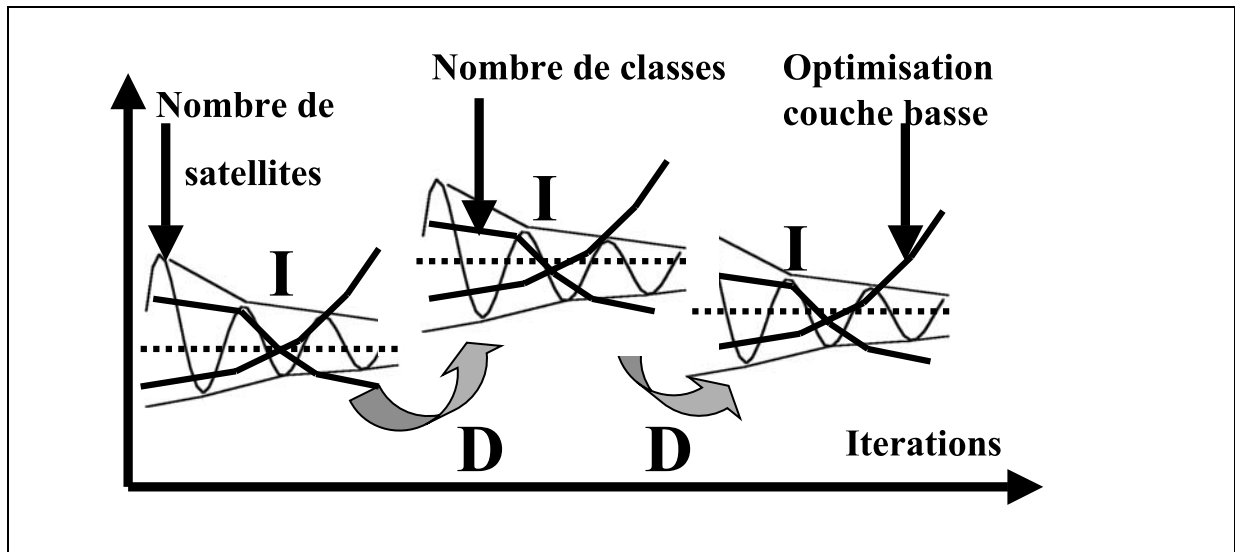


FIG. 13.7 – Les oscillations stratégiques

13.10.2 Évaluation de la progression d'une solution

Pour pouvoir décider si une diversification doit être mise en oeuvre, il est nécessaire d'évaluer dans quelle mesure la solution courante progresse. Cette évaluation fait intervenir deux paramètres : la valeur de la configuration (V_{cf}^1) et la composition de la configuration ($(N_{s_1}, \dots, N_{s_m})$).

Deux seuils (S_p^1 et S_p^2) sont fixés pour ces valeurs afin d'évaluer la progression de la configuration courante (Cf) par rapport à une configuration de référence (Cf_0). Si $V_{cf}(Cf) - V_{cf}(Cf_0) > S_p^1$ ou $d(Cf, Cf_0) > S_p^2$ alors on considère que la solution a progressé.

13.10.3 Que faire si le problème a été satisfait

Dans ce cas de figure, on a trouvé une solution (configuration Cf_1) au problème comportant un nombre n_1 de satellites. Deux stratégies peuvent être adoptées dans ce cas en fonction des objectifs généraux de la recherche. Si on cherche la première solution trouvée, l'algorithme s'arrête et renvoie la solution. Si on cherche à trouver une solution ayant un nombre minimal de satellites, alors on fixe la valeur de n_{min} et n_{max} de manière à ce que la majorité des valeurs possibles pour n soit inférieure à n_1 . On n'écarte pas toutes les solutions comportant plus de n_1 satellites car on autorise à passer momentanément par une configuration comportant plus de n_1 satellites (la recherche pouvant ensuite converger vers une configuration ayant moins de satellites). Par contre, la configuration Cf_1 est considérée comme borne inférieure tabou. C'est-à-dire que l'on écarte toutes les configurations Cf' telles que $Cf_1 \preceq Cf'$.

13.11 Conclusion

L'ensemble des aspects, standards et avancés, décrits dans ce chapitre font du niveau métaheuristique le véritable chef d'orchestre de l'algorithme. C'est un niveau complexe très fortement paramétré (par de nombreux seuils) qui s'attaque à un problème à forte combinatoire tant dans la taille de l'espace de recherche que dans la taille du voisinage des solutions explorées. Diverses informations remontent des niveaux inférieurs (évaluation des satellites, évaluation des classes), pour permettre de faire le lien entre ces différents niveaux. Les interactions sont donc horizontales (au sein du niveau métaheuristique) et verticales (entre les niveaux). Nous verrons dans la partie suivante des applications de cet algorithme dans le cadre des télécommunications et de la navigation. Ceci nous donnera l'occasion de donner des jeux de valeurs pour les différents paramètres présentés.

Intégration et conclusion

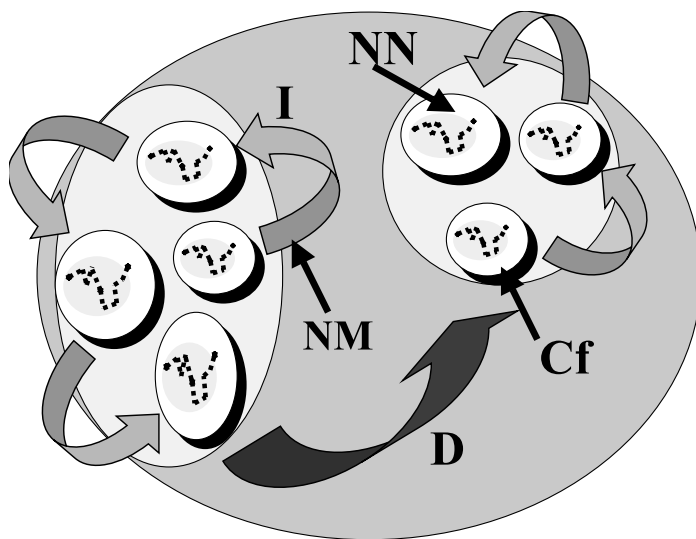


FIG. 13.8 – *Algorithme général*

Nous avons présenté au cours de cette partie l'ensemble des difficultés liées à la nature du problème ou aux algorithmes utilisés pour le résoudre. Au regard de chacun de ces points nous avons proposé une méthode de résolution nous conduisant à la spécification d'un algorithme à plusieurs niveaux. Nous allons maintenant récapituler l'enchaînement des différentes méthodes et l'imbrication des différents niveaux à l'aide de la figure 13.8 et de l'algorithme 13.2. On rappelle les notations adoptées au cours de cette partie :

- n_{min} : nombre minimal de satellites
- n_{max} : nombre maximal de satellites.
- H : histoire de l'algorithme (englobe

tous les niveaux de mémoire) (13.6)

- LT_{Cf}^1 : liste des k dernières configurations visitées (et donc potentiellement tabou) (13.6.1, 13.7.1).
- Δ : liste des configurations visitées (13.5.4)
- LT_{St} : liste des k' derniers satellites ajoutés (13.6.1).
- S_{cl} : seuil d'acceptation des classes (13.7.3)
- S_{cf} : seuil d'acceptation des configurations (13.7.3)
- S_{SD} : seuil de détection des maximums locaux de la recherche de bas niveau (12.5)
- N_{SD} : nombre d'itérations de la recherche de bas niveau (12.5)
- N_{os} : nombre d'itérations de la phase d'intensification (13.10)

TAB. 13.2 – *Algorithme général*

<p>Initialisation : $n_{min}, n_{max}, H, ODB, Cf$, Phase = Intensification <u>tantque</u> condition d'arrêt non satisfaite // Boucle de haut niveau <u>Si</u> Diversification <u>alors</u> Initialiser $N_{SD}, S_{SD}, n_{min}, n_{max}, S_{cl}$ <u>FinSi</u> <u>Faire</u> pendant N_{os} itérations // Boucle d'oscillation stratégique Calculer $V_{cf}^2(Cst')$ pour $Cf' \in V(Cf, H)$ (13.5.5) Construire la liste de candidats (13.8.2), en intégrant les critères d'aspiration (13.8.3) Choisir un candidat Cf^+ dans la liste (13.8.1) Ajouter Cf^+ à LT_{Cf}^1 (13.6.1). Ajouter Cf^+ à Δ (13.8.1) Retirer éventuellement une configuration de LT_{Cf}^1 (taille maximale atteinte) <u>Si</u> Cf^+ a déjà été visitée et que l'on est en diversification <u>alors</u> Réinitialiser les représentants avec les meilleures représentants de Cf^+ (13.9.4) <u>Sinon si</u> transition de type T_0 <u>alors</u> //(13.9.1) Initialiser les paramètres orbitaux du nouveau satellite (13.9.1) Ajouter le satellite à LT_{St} (13.6.1) Retirer éventuellement un satellite de LT_{St} (taille maximale atteinte) (13.6.1) Mettre à jour la fréquence d'utilisation de la classe associée au satellite (13.6.3) <u>Sinon si</u> transition de type T_1 <u>alors</u> //(13.9.2) Supprimer le représentant de plus faible évaluation (13.5.2, 13.6.2, 13.9.2) <u>Sinon</u> //transition de type T_2 (13.9.3) Sélectionner le représentant de plus faible évaluation (13.9.3) Changer sa classe d'affectation (13.9.3) <u>FinSi</u> Lancer le processus de recherche de bas niveau (algorithme 12.2) Mémoriser la meilleure constellation visitée, calculer la valeur V_{cf}^1 (13.6.4) <u>Si</u> Intensification <u>alors</u> Diminuer S_{SD}, augmenter N_{SD} (12.4,13.10) Augmenter S_{cl} (13.10). Adapter n_{min} et n_{max} (13.10) <u>FinSi</u> <u>Fin Faire</u> <u>Si</u> non progression de la solution <u>alors</u> (13.10.2) Phase = Diversification <u>FinSi</u> <u>fin tantque</u></p>

Partie V

Application

Nous allons présenter maintenant des cas d'application de l'algorithme présenté au cours de la partie précédente. Comme nous l'avons détaillé, l'algorithme est composé de deux niveaux de résolution : le niveau numérique et le niveau métaheuristique. Ces deux niveaux mettent en application des techniques d'optimisation différentes : un algorithme de recherche de plus grande pente avec une technique de redémarrage pour le niveau bas et un algorithme de recherche tabou avec des techniques avancées pour le niveau haut. Les différences entre les deux techniques sont présentes tant dans la manière d'optimiser la solution, que dans l'espace de recherche au travers duquel elles naviguent. Pour rendre compte des performances de chacune des parties de l'algorithme et ainsi justifier sa présence nous avons besoin de les évaluer séparément. Une évaluation globale ne permettrait pas, en effet, d'analyser finement les capacités de chacune des parties qui composent l'algorithme. Le premier chapitre (chapitre ??) est consacré à l'évaluation de la partie basse de l'algorithme à travers divers scénarios. Nous allons ensuite au cours de cette partie évaluer les performances de l'algorithme dans le domaine des télécommunications (chapitre 15) puis dans le domaine de la navigation (chapitre 16). Au cours de ces deux chapitres nous réaliserons une évaluation progressive en testant dans un premier temps le niveau bas de l'algorithme (en inhibant le niveau haut) puis nous testerons le niveau haut de l'algorithme (en inhibant le niveau bas) et enfin nous testerons l'algorithme dans sa globalité.

Par ailleurs, pour tester notre algorithme, nous devons définir des problèmes :

1. qui soient de natures, différentes, pour évaluer l'algorithme sous plusieurs angles,
2. pour lesquels on peut apprécier la qualité de la solution trouvée.
3. qui soient dignes d'intérêt pour la "communauté spatiale",

Pour répondre aux deux premiers points, nous allons étudier des problèmes déjà traités avec d'autres approches. Ces problèmes ont l'avantage d'être diversifiés et de posséder une (des) solution(s) connue(s) (dont l'optimalité n'est pas nécessairement prouvée), que nous pourrions comparer aux résultats fournis par notre algorithme. Pour satisfaire le troisième point, nous devrions considérer des problèmes non résolus émergeant d'appels d'offres récents. Dans le contexte économique actuel, peu d'études nouvelles sont proposées, nous allons donc créer artificiellement un problème faisant apparaître des dissymétries.

Chapitre 14

Évaluation de l'algorithme de bas niveau

Nous allons présenter maintenant les différents tests que nous avons réalisés sur la couche basse afin d'évaluer ses performances.

Les résultats sont regroupés dans les tableaux et graphiques 14.1 à 14.17.

Ces nombreux tests nous permettent d'apprécier les performances de l'algorithme en terme de temps de calcul, de convergence et d'optimalité. Ils nous permettent également de juger de l'utilité du redémarrage en complément de l'algorithme de descente.

14.1 Évaluation de l'algorithme de descente

14.1.1 Conditions des tests

L'algorithme est déterministe (pas de redémarrage). Les paramètres ayant une influence sur le déroulement de l'algorithme sont : le point initial, l'ensemble P_{SD} , la fonction critère et le voisinage. Les valeurs présentées dans les tableaux 14.1 et 14.2 sont *le temps de calcul / la valeur de l'optimum*. Les graphiques 14.3 à 14.6 représentent la valeur de l'optimum en fonction des itérations.

On fixe par ailleurs $P_{SD} = P_i$, $S_{SD} = 0$ (pas de redémarrage), et $N_{SD} = 1000$.

Afin de mieux se rendre compte de la durée des calculs, on précise que les tests ont été effectués sur un *PC Intel Pentium II 266 Mhz avec 128Mo de RAM*.

Pour la visibilité des satellites, l'angle d'élévation minimal est de 5 degrés.

Trois scénarios, ont été mis en place

- scénario 1 : couverture continue d'un point avec 1 satellite Toundra,
- scénario 2 : couverture continue de la bande de latitude $[-70,70]$ avec 3 puis 4 satellites quelconques,
- scénario 3 : couverture continue de la zone $[0,90] \times [0,90]$ avec 3 puis 4 satellites quelconques,

Les points initiaux sont fixés de la manière suivante.

Pour le scénario 1, le point initial est obtenu à partir de la solution optimale à laquelle on a appliqué une perturbation sur les paramètres P_{SD}

- P_0 : de 10 %
- P_1 : de 20 %
- P_2 : de 50 %
- P_3 : de 80 %

Pour les scénario 2 et 3

- P_0 : 3 satellites LEO, puis 4 satellites LEO
- P_1 : 3 satellites MEO, puis 4 satellites MEO
- P_2 : 2 satellites LEO et 1 satellite MEO, puis 2 satellites LEO et 2 satellites MEO.
- P_3 : 1 satellite LEO, 1 satellite MEO et 1 satellite GEO, puis 2 satellites LEO, 1 satellite MEO et 1 satellite GEO

Chacun de ces tests sera réalisé avec le voisinage complet et avec le voisinage incomplet.

14.1.2 Tableaux

NSD = 1000		PSD		
Tps / Val		a,i	a,i,e,w	a,i,e,w,W,M
Scénario 1	P0	2-29/91	4-17/85	21-16/48
1 satellite	P1	2-37/73	4-01/66'	21-17/45
Vmax=288	P2	2-35/50	4-21/41	21-05/52'
	P3	2-32/43	4-14/40	21-11/49'
Scénario 2	P0	16-15/280	32-18/315	47-55/362
3 Satellites	P1	16-38/1816	31-58/1756	48-17/1793
Vmax=2016	P2	16-15/784	32-22/724	48-31/637
	P3	16-56/1346	32-14/1278	48-29/1460
Scénario 2	P0	17-53/324	34-16/290	50-47/428
4 Satellites	P1	18-04/1912'	34-22/1953'	50-39/1887'
Vmax=2016	P2	18-11/956'	34-19/842'	50-16/913'
	P3	?	34-11/1314'	50-29/1411'
Scénario 3	P0	14-45/108	29-04/126	44-12/119
3 Satellites	P1	14-23/638	28-57/539	44-24/542
Vmax=900	P2	14-52/316	29-01/351	44-35/312
	P3	14-39/430	29-07/347	?
Scénario 3	P0	15-08/119	31-14/143	45-24/147
4 Satellites	P1	15-12/782	31-21/714	45-19/692
Vmax=900	P2	15-21/536	31-19/601	45-36/547
	P3	?	?	?

FIG. 14.1 – Test avec le voisinage incomplet

NSD = 1000		PSD		
Tps / Val		a,i	a,i,e,w	a,i,e,w,W,M
Scénario 1	P0	4-12/104	17-52/105	1h05/79
1 satellite	P1	4-16/85	17-23/92	1h05/68
Vmax=288	P2	4-27/83	17-39/86	1h04/72
	P3	04-09/71'	18-04/74'	1h05/70
Scénario 2	P0	30-04/343	8h10/348	2J/378
3 Satellites	P1	30-23/1834	8h10/1878	2J/1898
Vmax=2016	P2	30-12/823	8h09/927	?
	P3	30-32/1578	8h10/1498	?
Scénario 2	P0	35-12/326	~10h/378	?
4 Satellites	P1	35-16/1934	~10h/1967	?
Vmax=2016	P2	35-08/1045	~10h/1037	?
	P3	35-02/1886	~10h/1457	?
Scénario 3	P0	22-26/156	6h30/178	?
3 Satellites	P1	22-21/765	6h31/645	?
Vmax=900	P2	22-15/576	6h30/351	?
	P3	22-18/460	6h28/465	?
Scénario 3	P0	28-14/187	9h50/163	?
4 Satellites	P1	28-25/841	9h48/756	?
Vmax=900	P2	28-17/637	9h51/697	?
	P3	28-11/534	9h50/498	?

FIG. 14.2 – Test avec le voisinage complet

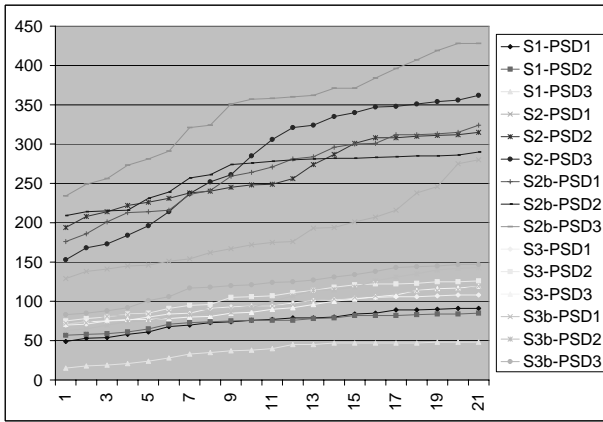


FIG. 14.3 – Valeur de l'optimum avec P_0

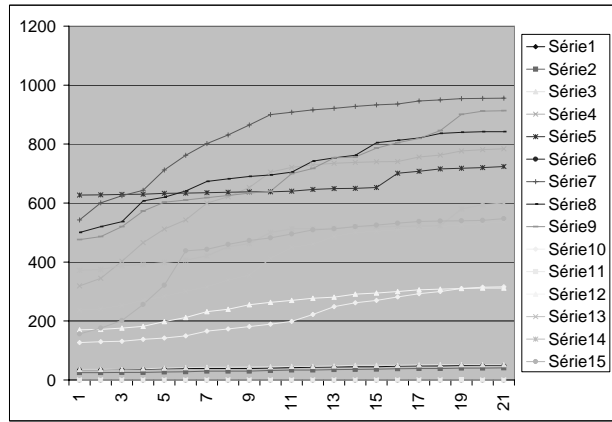


FIG. 14.5 – Valeur de l'optimum avec P_2

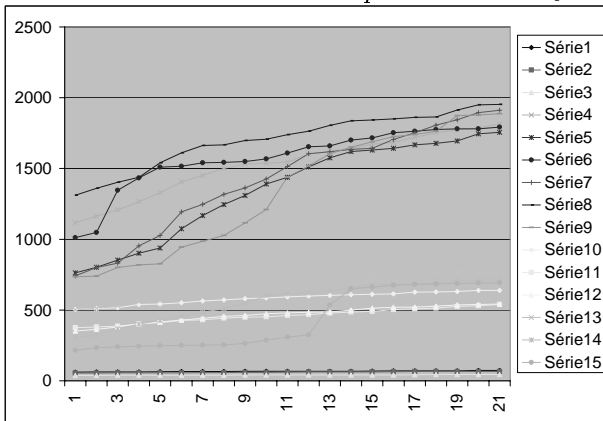


FIG. 14.4 – Valeur de l'optimum avec P_1

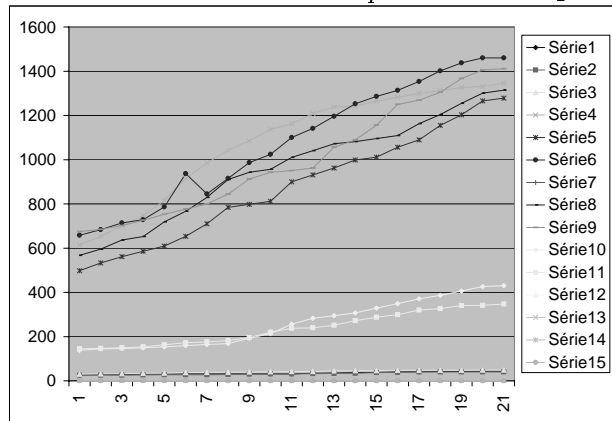


FIG. 14.6 – Valeur de l'optimum avec P_3

14.1.3 Interprétation

Temps de calcul

On constate, en comparant les trois colonnes du tableau 14.1, que le temps de calcul est proportionnel au nombre de paramètres utilisés pour l'algorithme de descente et ce quelque soit le scénario et le point initial (même constatation avec le tableau 14.2).

En comparant les tableaux 14.1 et 14.2 on constate également que le temps de calcul est proportionnel au voisinage :

- colonne 1, ligne 1 : tableau 14.1 : 2 mn 19 s, tableau 14.2 : 4 mn 12 s.
- colonne 2, ligne 3 : tableau 14.1 : 32 mn 22 s, tableau 14.2 : 8 h 09 mn.

Les temps de calculs associés au voisinage complet sont très long et on arrive aux limites acceptables lorsque le nombre de paramètres P_{SD} et le nombre de satellites augmentent :

- colonne 3, lignes 1 et 2 : 2 jours de calcul
- colonne 3, lignes 7 à 20 : temps de calcul trop important

Convergence

Au niveau de la convergence, les graphiques 14.3 à 14.6 montrent une croissance continue et une bonne convergence vers un optimum local. Cet optimum local est un piège pour la recherche de l'optimum global mais ce n'est pas l'objectif de cette partie de l'algorithme que de le détecter, on se reporte pour cela au redémarrage évalué par la suite. En outre la convergence est relativement lente et des techniques d'accélération peuvent être mise en place.

Optimalité

Si on s'intéresse maintenant à l'optimalité des solutions du point de vue spatial, on constate que :

- l'algorithme est sensible aux grandes tendances. Les paramètres dont un changement de valeur a des répercussions immédiates sur la valeur des solutions sont rapidement déterminés et modifiés. C'est le cas de l'augmentation de l'altitude des satellites qui permet d'améliorer les résultats. C'est également le cas avec le phasage des satellites qui permet d'éviter les redondances.
- l'algorithme semble avoir du mal à synchroniser des orbites basses. Ceci est certainement lié au fait que les satellites LEO défilent rapidement dans le ciel de l'utilisateur et qu'il faut donc beaucoup plus de satellites pour satisfaire les besoins. La synchronisation de ces derniers est donc primordiale.
- Dans le cas mono-satellite (tableau 14.1 lignes 1 à 4) et dans le cas de satellites géostationnaires (point initial P_3 des scénarios 2 et 3) l'algorithme permet d'obtenir un taux de couverture tout à fait satisfaisant compte tenu du nombre de satellites présents.

14.2 Évaluation du redémarrage

14.2.1 Conditions des tests

On teste ici l'algorithme complet (avec redémarrage aléatoire), on réalise 5 tests par scénario et on utilise uniquement le voisinage restreint.

Les tableaux 14.7 à 14.11 représentent *la valeur de l'optimum / le nombre de redémarrages* moyenne.

On fixe $S_{SD} = 0.5\%$, $N_{SD} = 1000$. On utilise le même ordinateur que pour le test précédent.

Les trois scénarios testés sont identiques aux précédents et les points initiaux sont fixés de la manière suivante.

Pour le scénario 1 on utilise le point P_0 des tests précédents et pour les scénarios 2 et 3 le point P_2 (avec 3 puis 4 satellites).

14.2.2 Tableaux

14.2.3 Interprétation

Le redémarrage est plus efficace lorsqu'il y a peu de paramètres (Tableau 14.9 : la valeur de l'optimum est maximale pour $P_{SD} = (a, i)$ (colonne 1) sauf pour $P_r = M$). Ces résultats confirment la convergence très douce (plus il y a de paramètres moins la convergence est rapide).

Le redémarrage permet, sous certaines conditions portant sur les paramètres P_r (peu de paramètres), d'augmenter la valeur de l'optimum (à nombre d'itérations N_{SD} égal par rapport aux tests précédents).

- Tableau 14.1 ligne 7, colonnes 1 à 3 : la valeur de l'optimum est respectivement : 784, 724 et 637.
- Tableau 14.8 lignes 1 à 5 et colonnes 1 à 3 : une seule valeur est inférieure à 784 (763 colonne 1, ligne 5). La moyenne étant de 822.

On constate également que le nombre de redémarrages dépend du nombre de paramètres P_{SD} . Plus il y a de paramètres, moins il y a de redémarrages.

L'algorithme de descente permet donc de converger vers un optimum local, et le redémarrage de changer de zone d'influence. Si cette association permet d'améliorer les résultats, les conditions initiales semblent avoir plus d'influence sur la solution trouvée que l'algorithme de descente.

14.3 Tests approfondis

14.3.1 Conditions des tests

Certains tests sur les scénarios 2 et 3 effectués précédemment méritent d'être poussés afin de mieux apprécier leurs limites.

Nous allons par exemple ajouter l'ensemble (Ω, M) aux ensembles P_{SD} testés. On relâche également le nombre d'itération N_{SD} précédemment limité à 1000 itérations. Dans le cas de l'algorithme de descente seul, on arrête le test lorsqu'il y a stabilité de la valeur du maximum (maximum local atteint). Dans le cas avec redémarrage, on se limite au nombre d'itérations maximal atteint précédemment afin de pouvoir apprécier l'apport du redémarrage.

On teste ici uniquement le point initial P_1 avec trois satellites. Dans le cas où l'optimum ne serait pas atteint avec trois satellites, on ajoute un quatrième satellite.

14.3.2 Tableaux

Les tableaux 14.12, 14.13 présentent respectivement les tests de l'algorithme de descente seul appliqué aux scénarios 2 et 3. Les tableaux 14.14 à 14.17 présentent les tests de l'algorithme de descente avec redémarrage avec 3 et 4 satellites selon le principe présenté précédemment.

14.3.3 Interprétation

En poussant la convergence on constate, tableau 14.12, que la valeur de l'optimum augmente (colonne 4, ligne 1: 1977 contre 1816 dans le test précédent tableau 14.1 colonne 1, ligne 6). Mais elle ne permet pas d'obtenir le maximum (2016) avec 3 satellites. On ajoute donc un satellite et on constate que, d'une part la valeur de l'optimum augmente (ce qui était prévisible) et que le maximum est atteint colonne 1, ligne 2 et colonne 4 ligne 2. Ces deux cas correspondent à des ensembles P_{SD} réduit à deux paramètres.

Dans le cas du scénario 3, tableau 14.13, la poursuite des itérations permet d'obtenir le maximum (900) dans 3 cas sur 4 avec 3 satellites (ligne 1 colonne 1, 2 et 4). Dans le cas où P_{SD} est complet (6 paramètres) l'ajout d'un quatrième satellite ne permet pas d'obtenir le maximum. On est dans un cas où la combinatoire est grande et la convergence est très lente ce qui explique peut être que malgré les 5600 itérations le maximum (même local) ne soit pas atteint.

Concernant l'évaluation du redémarrage, on constate que certains couples (P_{SD}, P_r) permettent d'obtenir le maximum avec 3 satellites (scénario 2 et 3). C'est notamment le cas pour $P_{SD} = (a, i)$ ou $P_{SD} = (a, i, e, \omega)$ et $P_r = (\Omega, M)$ (ligne 2, colonnes 1 et 2 des tableaux 14.14 et 14.16).

De manière générale, des ensembles P_{SD} et P_r trop grand ne permettent pas de converger raisonnablement vers un optimum.

- colonne 1, ligne 5 des tableaux 14.14 et 14.16): pas de convergence et P_r maximal.
- colonne 3 des tableaux 14.14 à 14.17: pas de convergence et P_{SD} maximal.

D'autre part, l'appartenance de a et i à P_{SD} semble importante de même que l'appartenance de Ω et M à P_r . L'ajout de a, e, i, ω à P_r n'apporte rien, elle n'engendre qu'une augmentation du nombre d'itérations avant convergence (scénario 2 tableau 14.14 colonne 1)

- 1826 itérations pour $P_r = (\Omega, M)$,
- 2134 itérations pour $P_r = (\Omega, M, a)$,
- 2468 itérations pour $P_r = (\Omega, M, a, i)$,
- 3618 itérations pour $P_r = (\Omega, M, \omega, e, a, i)$ avec 4 satellites (tableau 14.15).

Les mêmes constatations sont valables avec le scénario 3 et les tableaux 14.16 et 14.17.

Scénario 1 1 Satellite		PSD		
		a,i	a,i,e,w	a,i,e,w,W,M
Vmax=288	M	97/53	78/31	85/14
	W, M	106/47	94/41	92/18
Pr	a, W, M	99/38	87/37	97/12
	a,i,W,M	110/58	89/45	75/21
	a,e,i,w,W,M	88/51	92/36	81/16

FIG. 14.7 – Scénario 1, P_0 , 1 satellite

Scénario 2 3 Satellites		PSD		
		a,i	a,i,e,w	a,i,e,w,W,M
Vmax=2016	M	816/72	865/46	793/21
	W, M	819/75	799/53	812/19
Pr	a, W, M	856/67	875/27	801/13
	a,i,W,M	834/63	836/33	785/17
	a,e,i,w,W,M	763/54	879/37	799/24

FIG. 14.8 – Scénario 2, P_2 , 3 satellites

Scénario 2 4 Satellites		PSD		
		a,i	a,i,e,w	a,i,e,w,W,M
Vmax=2016	M	934/65	894/30	925/17
	W, M	990/56	879/34	938/23
Pr	a, W, M	978/79	899/37	910/35
	a,i,W,M	964/64	914/41	943/15
	a,e,i,w,W,M	987/58	885/33	934/17

FIG. 14.9 – Scénario 2, P_2 , 4 satellites

Scénario 3 3 Satellites		PSD		
		a,i	a,i,e,w	a,i,e,w,W,M
Vmax=900	M	310/110	342/46	314/32
	W, M	324/76	351/39	301/18
Pr	a, W, M	356/57	365/35	327/24
	a,i,W,M	351/61	379/29	334/12
	a,e,i,w,W,M	345/92	353/41	325/27

FIG. 14.10 – Scénario 3, P_2 , 3 satellites

Scénario 3 4 Satellites		PSD		
		a,i	a,i,e,w	a,i,e,w,W,M
Vmax=900	M	548/76	580/37	482/16
	W, M	579/57	622/35	559/9
Pr	a, W, M	550/46	657/31	501/16
	a,i,W,M	539/69	614/28	515/21
	a,e,i,w,W,M	528/67	639/40	500/13

FIG. 14.11 – Scénario 3, P_2 , 4 satellites

Vmax=2016		PSD		
		a,i	a,i,e,w	a,i,e,w,W,M
3 Satellites	P1	1700/1934	2100/1898	3050/1907
4 Satellites	P1	1467/2016	3300/1947	5400/1939

FIG. 14.12 – Scénario 2, P_1 , SD Seule

Vmax=900		PSD		
		a,i	a,i,e,w	a,i,e,w,W,M
3 Satellites	P4	1245/900	3750/900	5600/658
4 Satellites	P4	-	-	5600/823

FIG. 14.13 – Scénario 3, P_1 , SD seule

Scénario 2 3 Satellites		PSD			
		a,i	a,i,e,w	a,i,e,w,W,M	W,M
Vmax=2016	M	5400/1963/105	5400/1901/48	5400/1856/65	5400/1834/165
	W, M	1826/2016/75	2678/2016/103	5400/1879/93	5400/1821/214
Pr	a, W, M	2134/2016/114	5400/1986/104	5400/1901/86	5400/1917/137
	a,i,W,M	2468/2016/86	5400/1934/96	5400/1860/78	5400/1945/128
	a,e,i,w,W,M	5400/1835/143	5400/1913/85	5400/1894/67	5400/1896/132

FIG. 14.14 – Scénario 2, P_1 , 3 satellites

Scénario 2 4 Satellites		PSD			
		a,i	a,i,e,w	a,i,e,w,W,M	W,M
Vmax=2016	M	5400/1966/128	5400/1987/132	5400/1885/69	5400/1902/178
	W, M	-	-	5400/1948/73	5400/1892/197
Pr	a, W, M	-	3967/2016/96	5400/1973/67	5400/1911/186
	a,i,W,M	-	4136/2016/112	5400/1956/63	3839/2016/265
	a,e,i,w,W,M	3618/2016/143	4915/2016/101	5400/1965/76	4367/2016/244

FIG. 14.15 – Scénario 2, P_1 , 4 satellites

Scénario 3 3 Satellites		PSD			
		a,i	a,i,e,w	a,i,e,w,W,M	W,M
Vmax=900	M	1969/900/101	4154/900/69	5600/736/37	1166/900/124
	W, M	2951/900/89	4585/900/73	5600/728/48	1241/900/112
Pr	a, W, M	3235/900/63	5528/900/101	5600/701/36	5600/325/254
	a,i,W,M	3794/900/95	5600/848/98	5600/694/38	5600/389/239
	a,e,i,w,W,M	5600/768/108	5600/816/64	5600/673/33	5600/137/293

FIG. 14.16 – Scénario 3, P_1 , 3 satellites

Scénario 3 4 Satellites		PSD			
		a,i	a,i,e,w	a,i,e,w,W,M	W,M
Vmax=900	M	-	-	5600/827/41	-
	W, M	-	-	5600/846/43	-
Pr	a, W, M	-	-	5600/849/41	5600/436/261
	a,i,W,M	-	4978/900/156	5600/837/49	5600/356/270
	a,e,i,w,W,M	5139/900/148	5600/837/132	5600/845/63	5600/278/256

FIG. 14.17 – Scénario 3, P_1 , 4 satellites

Chapitre 15

Application en télécommunications

15.1 Introduction

Nous allons appliquer, dans ce premier chapitre, l'algorithme présenté dans la partie précédente à un problème de télécommunications. Nous commencerons par définir la manière dont nous allons évaluer les performances de l'algorithme (section 15.2). Puis nous effectuerons les tests en détaillant chacune des étapes de validation de l'algorithme : validation de la recherche de bas niveau, validation de la recherche de haut niveau, validation de l'algorithme global. Deux problèmes vont être traités au cours de ce chapitre. Nous allons calibrer l'algorithme sur un problème très simple, dont la solution (connue) ne comporte que 3 satellites (section 15.3).

15.2 Remarques préliminaires sur les techniques d'évaluation

Nous allons donner les idées qui nous ont guidées dans l'élaboration des jeux de tests présentés dans ce chapitre ainsi que dans le chapitre suivant.

15.2.1 Test du niveau numérique

Pour tester le niveau numérique il faut annuler l'influence du niveau métaheuristique. Pour cela nous allons fixer la configuration dans laquelle la recherche de bas niveau va s'effectuer et analyser le parcours suivi par celle-ci.

Deux types de configurations vont être testés : des configurations contenant la solution et d'autres ne contenant pas la solution. Si l'objectif des tests réalisés dans le premier cas est de trouver la solution optimale, celui du deuxième cas est d'évaluer la capacité de l'algorithme de bas niveau à mailler la configuration en faisant émerger les meilleures solutions qu'elle contient afin de lui attribuer une note représentative.

Il faut par ailleurs noter que l'assertion "*la configuration contient la solution*" n'a de valeur qu'en liaison avec une certaine définition des classes d'orbites intégrées dans la base de données *ODB*. Typiquement, considérons une unique classe d'orbites ne comportant aucune restriction par rapport aux plages de variation maximales des paramètres orbitaux ($a \in [400km, 36000km], e \in [0, 0.9], i \in [0, 360], \omega \in [0, 360], \Omega \in [0, 360], M \in [0, 360]$). L'assertion "*la solution appartient à la configuration* $Cf = \{n\}$ " n'apporte alors que la seule information du nombre de satellites de la constellation optimale, les paramètres orbitaux pouvant avoir quant à eux n'importe quelle valeur. Inversement, si on considère que toutes les classes d'orbite Cl_k n'admettent qu'un seul jeu de paramètres orbitaux ($Cl_k = \{a_k, e_k, i_k, \omega_k, \Omega_k, M_k\}$) alors la donnée de la configuration suffit à définir précisément la constellation. Entre les deux extrêmes les possibilités sont nombreuses et nous verrons que la recherche d'une solution optimale au sein d'une configuration peut prendre toutes les tournures. Nous réaliserons différents tests avec des tailles et formes de classes différentes pour mesurer leur impact sur la durée de la recherche et la qualité de la solution trouvée.

15.2.2 Test du niveau métaheuristique

A l'opposé des tests précédents, nous voulons ici tester le niveau haut de l'algorithme, autrement dit nous devons abstraire la recherche de bas niveau. Pour annuler l'influence de l'optimisation au sein d'une classe, nous devons nous assurer qu'après être sortis d'une configuration, la meilleure solution, accessible dans la configuration explorée, ait été trouvée.

Par ailleurs nous veillerons à évaluer la qualité de l'évaluation à priori d'une configuration afin de justifier le bien-fondé de la formule employée.

Enfin nous contrôlerons le bon fonctionnement des différentes composantes de ce niveau (oscillations stratégiques, application des transitions, ...).

15.2.3 Test de l'algorithme global

Nous intégrons ensuite les deux niveaux, les tests vont consister à évaluer différents jeux de paramètres conduisant à des compromis entre le niveau bas et le niveau haut. Les tests seront là encore progressifs dans la mesure où l'algorithme aura à faire face à toutes les difficultés (optimisation de bas niveau, optimisation de haut niveau).

15.3 Recherche du nombre minimal de géostationnaires

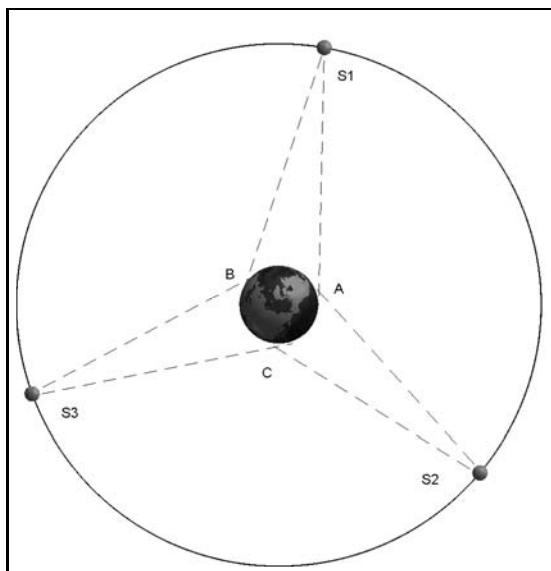


FIG. 15.1 – *Solution 3 GEO*

Le problème proposé ici est très simple, il s'agit de déterminer le nombre minimal de satellites géostationnaires pour couvrir la bande de latitude $[-65, +65]$. La solution à ce problème est connue et comporte trois satellites (S_1, S_2, S_3 sur la figure 15.1) géostationnaires positionnés comme sur la figure 15.1, c'est-à-dire une répartition uniforme des satellites autour de l'équateur. Mais le recouvrement partiel des cercles de visibilité des trois satellites (zones A, B et C) n'assure pas l'unicité de la solution. Chaque satellite peut en effet bouger autour de la position indiquée précédemment. Pour toutes ces solutions, la couverture de la bande $[-65, +65]$ est totale et on considérera que toute solution satisfaisant cette propriété sera déclarée comme valide.

Nous allons utiliser au cours de notre algorithme :

- La fonction d'évaluation d'une constellation f renvoie le nombre total de points couverts. On cherche donc le jeu de paramètres orbitaux qui maximise f , et plus précisément la constellation $C_{s_{opt}}$ telle que $f(C_{s_{opt}}) = NbPt$, $NbPt$ étant le nombre total de points de maillage de la bande de latitude $[-65, +65]$.
- On cherche à minimiser le nombre de satellites.
- La fonction d'évaluation d'un satellite est : $V_{st} = V_{s_1}$ (chapitre 13 page 165). On cherche à maximiser cette valeur, on favorise donc les satellites qui couvrent des zones sans redondance avec un autre satellite (compatible avec la minimisation du nombre de satellites). Au lieu de maximiser la valeur de V_{s_1} , nous aurions pu minimiser la valeur de V_{s_2} (qui comptabilise les zones couvertes de manière redondante) car étant donné que les orbites considérées sont toutes géostationnaires, le nombre total N de points couverts durant la simulation par un satellite est commun à tous les satellites (même orbite donc même taille de cercle de visibilité et de plus la zone est uniforme vis-à-vis de l'équateur). Or $N = V_{s_1} + V_{s_2}$ donc maximiser V_{s_1} est équivalent à minimiser V_{s_2} .

15.3.1 Test du niveau numérique

Hypothèses

Pour tester ce niveau nous considérons que la configuration est fixée. Par ailleurs nous ne considérons qu'une seule classe dans l'*ODB* : la classe Géostationnaire dont les paramètres sont :

- $If_a = \{Re + 36000km, Re + 36000km, Re + 36000km, Re + 36000km\}$
- $If_e = \{0,0,0,0\}$
- $If_i = \{0,0,0,0\}$
- $If_\omega = \{0,0,0,0\}$
- $If_\Omega = \{0,0,0,0\}$
- $If_M = \{0,0,360,360\}$

Nous analyserons la recherche au sein des trois configurations suivantes :

- $Cf_3 = \{3\}$ (une seule classe et 3 satellites)
- $Cf_4 = \{4\}$ (une seule classe et 4 satellites)
- $Cf_2 = \{2\}$ (une seule classe et 2 satellites)

Les recherches seront menées jusqu'à leur terme (convergence vers une solution de valeur maximale), les paramètres N_{SD} et S_{SD} définis dans le chapitre 12 n'interviennent donc pas dans ce test.

Optimisation dans la configuration Cf_3

La solution appartient à cette configuration et nous allons vérifier que l'algorithme converge bien vers celle-ci.

La position initiale des satellites est fixée de manière aléatoire. On constate que quelle que soit la valeur initiale des paramètres, l'algorithme converge très rapidement. Cette rapidité est liée au fait que la fonction d'évaluation ne possède pas de maximum local. L'algorithme de plus forte pente converge en ayant déterminé uniquement deux directions de recherche (figure 15.2). D'après l'algorithme décrit dans le chapitre 12, lorsqu'une direction de descente est détectée, l'algorithme continue dans la même direction tant que le critère f augmente. Dans de telles conditions les seuls paramètres qui influent sur la rapidité de convergence sont : la valeur des paramètres orbitaux initiaux (plus ils sont éloignés et plus la convergence sera longue); le pas de discrétisation du paramètre M . Dans l'exemple de la figure 15.2, le pas est de 1 degré (la position des satellites est donnée par la figure) et conduit à une convergence en :

- 2 directions de recherche au total (6 possibles à chaque itération).
- 98 pas.
- 14 secondes (Pentium III 600 Mhz).

Le temps de calcul est particulièrement court car la simulation est réduite du fait de la stabilité d'un satellite géostationnaire (dans le ciel de l'utilisateur) . On constate par ailleurs (figure 15.3) que la valeur de la constellation augmente linéairement (Chaque itération permet au satellite modifié d'ajouter un même nombre de points couverts de manière non redondante).

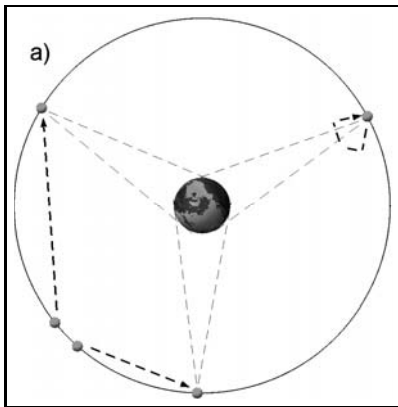


FIG. 15.2 – *Solution trouvée*

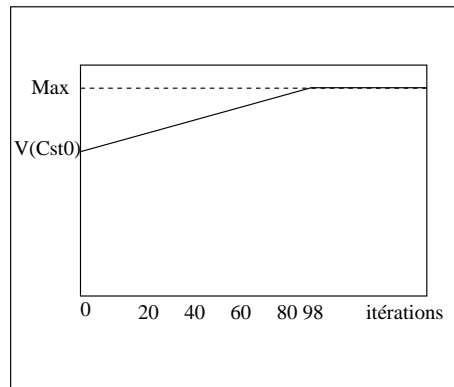


FIG. 15.3 – *Valeur de la constellation*

Optimisation dans la configuration Cf_4

La configuration Cf_4 comporte un satellite de plus que le nombre requis, elle permet donc de satisfaire le problème de couverture mais pas en nombre minimal de satellites. Au niveau de la recherche, il n'y a toujours pas d'optimum local mais un grand nombre de solutions consécutives de valeur maximale (redondance de couverture). L'exploration de la configuration permet d'obtenir une solution de valeur maximale de manière très rapide.

Optimisation dans la configuration Cf_2

Contrairement aux deux configurations précédentes, Cf_2 ne contient pas de solution. Mais pour l'algorithme de bas niveau, qui a pour objectif de trouver la meilleure constellation compatible avec la configuration en cours, cette donnée n'a pas d'influence. Il existe comme dans le cas précédent beaucoup d'optimum globaux (au sein de Cf_2 , qui sont des optimums locaux si on se ramène au problème général) qui sont obtenus lorsque les deux satellites n'ont plus de points en visibilité commune ($V_{S_2} = 0$ pour chaque satellite). Une seule direction de descente est nécessaire et l'algorithme converge en quelques itérations.

15.3.2 Test du niveau métaheuristique

Hypothèses

Nous allons tester maintenant le fonctionnement du niveau métaheuristique. Pour valider ce niveau, il faut s'assurer que l'algorithme de recherche tabou permet de naviguer correctement dans l'espace des configurations. Si nous considérons que la base *ODB* est composée uniquement de la classe géostationnaire (section 15.3.1) et que l'on inhibe le niveau numérique (section 15.2.2) en renvoyant systématiquement au terme de l'exploration d'une configuration, la meilleure constellation accessible via celle-ci, alors, la recherche a peu d'intérêt puisqu'il n'y a que $n_{max} - n_{min}$ configurations possibles (donc $n_{max} - n_{min}$ tests à effectuer). Nous devons disposer d'un nombre suffisant de configurations pour que la recherche de niveau métaheuristique ait un sens. Nous allons donc créer artificiellement un certain nombre de sous-classes de la classe géostationnaire afin d'augmenter la complexité de l'espace de recherche.

On définit ainsi 6 sous classes de la manière décrite par la figure 15.4. On constate que les classes sont contiguës : les intervalles flous sur le paramètre *M* sont choisis de manière à ce que les extrémités soient communes à deux classes (figure 15.5). Par ailleurs, on pose $n_{min} = 1$ et $n_{max} = 5$, il y a donc 9330 combinaisons possibles (équation 13.2, page 162).

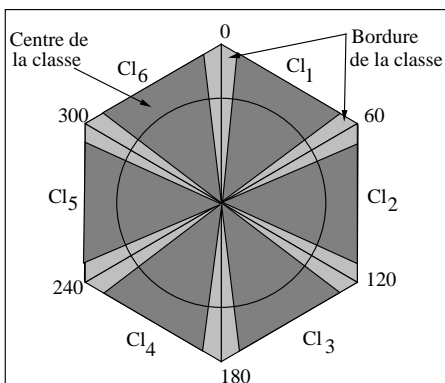


FIG. 15.4 – Définition des classes a)

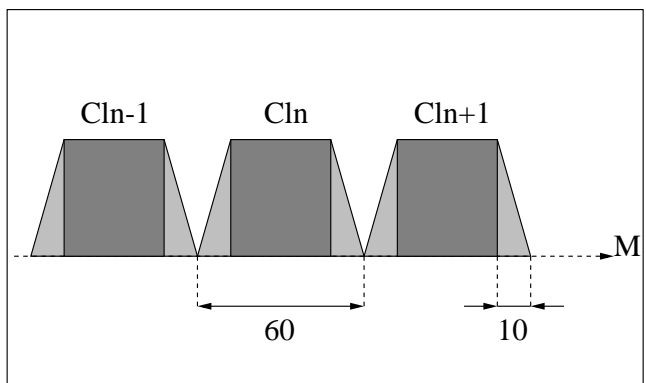


FIG. 15.5 – Définition des classes b)

Le processus d'optimisation de bas niveau est adapté pour ce cas d'application et renvoie une solution d'évaluation maximale pour chacune des configurations testées (cela revient à considérer un processus idéal qui n'influence pas le processus d'optimisation de haut niveau). Pour renvoyer une telle solution, nous procédons de la manière suivante : on essaye d'avoir la répartition la plus uniforme possible des satellites autour de l'équateur. Cette répartition tient compte du nombre de satellites ainsi que des classes d'appartenance qui définissent les plages de variation des différents paramètres.

Nous rappelons par ailleurs, que l'algorithme de haut niveau tend à intégrer dans la configuration courante les classes les plus susceptibles de répondre au problème. Ce choix est réalisé relativement à la valeur des classes (V_{cl}). Cette valeur est attribuée en fonction des configurations parcourues (Δ), et de la valeur *à posteriori* de celles-ci (V_{cf}^1). Or la valeur V_{cf}^1 est calculée en fonction du parcours de l'algorithme de bas niveau et de la valeur des constellations rencontrées ($f(Cs)$). Dans notre cas, la seule constellation rencontrée est la meilleure constellation accessible avec la configuration Cf ($Cs_{opt}(Cf)$): on a donc $V_{cf}^1 = f(Cs_{opt}(Cf))$.

La fonction $f(Cst)$ renvoie le nombre de points couverts par Cst divisé par $NbPt$ (valeur comprise entre 0 et 1). La valeur des classes est initialisée à 0.5. Les autres paramètres sont fixés de la manière suivante : $d_{max} = 2$, $N_{os} = 10$, $S_p^1 = 0.2$ (seuil de progression de la valeur de la configuration), $S_p^2 = 2$ (seuil de progression de la nature de la configuration) (chapitre 13 section 13.10.2).

Déroulement hypothétique de l'algorithme

Avant de présenter des résultats numériques, nous allons illustrer le principe de fonctionnement de l'algorithme en présentant un cas certes hypothétique mais décrivant un comportement guidé par les probabilités de sélection des divers candidats. Au cours de la présentation des résultats, nous allons particulariser le scénario de résolution en choisissant arbitrairement la première classe intégrée dans la configuration. Un autre choix conduirait à un déroulement similaire de l'algorithme. L'algorithme permet de résoudre le problème de la manière suivante

1. La configuration initiale est : $Cf_0 = \{0,0,0,0,0,0\}$. Pour le choix du successeur, nous avons besoin de définir une valeur *à priori* pour l'ensemble des configurations voisines de Cf_0 . Il y en a 6 notées Cf_i^+ , i étant la classe d'appartenance du représentant ajouté à Cf_0 pour obtenir Cf_i^+ . Sous ces conditions on pose $V_{cf}^2(Cf_i^+) = V_{cl}(Cl_i)$. Le voisinage est donc uniforme et chaque valeur *à priori* vaut 0.5. La première transition, nécessairement de type T_0 , est donc choisie aléatoirement.
2. On suppose que la transition sélectionnée est T_0^1 (on ajoute donc un représentant de Cl_1) et on obtient $Cf_1 = \{1,0,0,0,0,0\}$.
3. La meilleure constellation renvoyée par la version adaptée de l'algorithme de bas niveau est Cst_1 (composée d'un satellite) dont la valeur M est 30 (section 13.9.1, chapitre 13).
4. On obtient donc : $f(Cst_1) = V_{cf}^1(Cf_1) = V_{cl}(Cl_1) = 1/3$, (section 13.5.5 chapitre 13)
5. Choix du successeur de Cf_1 , voisinage de Cf_1 : $V(Cf_1, H) = \{(2,0,0,0,0,0), (1,1,0,0,0,0), (1,0,1,0,0,0), (1,0,0,1,0,0), (1,0,0,0,1,0), (1,0,0,0,0,1), (0,1,0,0,0,0), (0,0,1,0,0,0), (0,0,0,1,0,0), (0,0,0,0,1,0), (0,0,0,0,0,1)\}$, dont les valeurs *à priori* sont $\{4,5,5,5,5,5,6,6,6,6,6\}/12$.
6. Les valeurs *à priori* sont très proches, mais la priorité est donnée sur les transitions de type T_2 . En pratique, l'algorithme explore les configurations comportant 1 satellite en changeant

successivement la classe d'appartenance. Après ces 5 tentatives infructueuses, la liste des configurations tabou (les dernières configurations visitées) pousse l'algorithme à visiter une configuration à 2 satellites. On considère que la configuration sélectionnée est $Cf_2 = \{1,1,0,0,0\}$.

7. La meilleure constellation Cst_2 , renvoyée par l'algorithme de bas niveau est telle que $M = 0$ pour le représentant de Cl_1 et $M = 120$ pour le représentant de Cl_2 . Les valeurs sont $f(Cst_2) = V_{cf}^1(Cf_2) = 2/3$, $V_{cl}(Cl_1) = V_{cl}(Cl_2) = 1/2$.
8. Le voisinage de la configuration Cf_2 se trouve composé de configurations à 1,2, ou 3 satellites. Parmi elles, les configurations $\{2,1,0,0,0\}$ et $\{1,2,0,0,0\}$ ont la plus grande évaluation *a priori*, du fait de la valeur des classes Cl_1 et Cl_2 .
9. Statistiquement, le choix conduit vers l'une ou l'autre des configurations. Supposons que la prochaine configuration visitée soit $Cf_3 = \{1,2,0,0,0\}$, la meilleure constellation Cst_3 renvoyée par le niveau bas est alors telle que $M = 0$ pour le représentant de Cl_1 , $M = 60$ pour le premier représentant de Cl_2 et $M = 120$ pour le second représentant de Cl_2 .
10. Les valeurs obtenues sont : $f(Cst_3) = V_{cf}^1(Cf_3) = f(Cst_2) = 2/3$, $V_{cl}(Cl_1) = 5/9$, $V_{cl}(Cl_2) = 2/3$. Par ailleurs, $V_{s1} = 0$ pour le premier représentant de Cl_2 . Autrement dit, ce satellite est redondant et les critères d'aspirations (chapitre 13 section 13.8.3) conduisent à la suppression de celui-ci.
11. On obtient donc la configuration Cf_4 identique à Cf_2 .
12. Une alternative au choix réalisé à la configuration Cf_2 est par exemple d'explorer la configuration $Cf_5 = \{1,1,0,0,1\}$ avec $M = 300$ pour le représentant de Cl_6 , $M = 30$ pour celui de Cl_1 et $M = 120$ pour celui de Cl_2 dans la constellation optimale Cst_5 .
13. On obtient : $f(Cst_5) = V_{cf}^1(Cf_5) = 5/6$ et $V_{cl}(Cl_1) = 19/30$, $V_{cl}(Cl_2) = 21/30$, $V_{cl}(Cl_6) = 25/30$.
14. Statistiquement, la sélection du successeur de la configuration Cf_5 doit conduire à explorer le même type de configuration que Cf_3 (c'est-à-dire comportant un satellite redondant). Suite à cette exploration, l'algorithme revient à la configuration $Cf_6 = Cf_5$.
15. A ce niveau, une autre alternative est d'appliquer une transition de type T_2 sur le représentant de la classe Cl_6 . D'après la section 13.9.3 du chapitre 13, la classe de remplacement est la classe Cl_5 .
16. Sous cette condition, une constellation optimale émerge ($M = 0$ pour le représentant de Cl_1 , $M = 120$ pour le représentant de Cl_2 , $M = 240$ pour le représentant de Cl_5).

Analyse des résultats

L'algorithme fonctionne de manière très efficace grâce à la continuité des classes. Cela illustre l'utilité de la classification qui permet de guider la recherche. En créant un ensemble de classes

contiguës, on définit un parcours potentiel pour les représentants des classes. Cette évolution se traduit au niveau des configurations par l'émergence de chaînes de configurations dont la continuité est assurée par application du type de transition T_2 . Pour fixer un peu plus les idées, nous allons considérer le cas suivant :

- Les classes Cl_1 , Cl_2 et Cl_3 sont telles que $Cl_1 \cap Cl_2 \neq \emptyset$, et $Cl_2 \cap Cl_3 \neq \emptyset$.
- La configuration courante est $Cf = \{3,0,0\}$

Si on considère qu'après l'exploration de Cf la meilleure constellation trouvée est telle que l'un des représentants de Cl_1 appartienne à $Cl_1 \cap Cl_2$, alors le successeur de Cf par application d'une transition de type T_2 est $Cf^+ = \{2,1,0\}$ (on change la classe d'appartenance, en prenant la classe la plus proche, section 13.9.3, chapitre 13). De cette nouvelle configuration, deux autres sont alors contiguës ($\{2,0,1\}$ et $\{1,2,0\}$) et ainsi de suite. On définit ainsi une chaîne de configurations qui assure une continuité dans l'optimisation des paramètres orbitaux au niveau le plus bas.

Il faut également rappeler que le recouvrement partiel de classes est possible (section 11.11, chapitre 11) et permet de guider de manière encore plus directe la succession des classes d'appartenance d'un représentant. Mais cette dernière technique est à utiliser avec modération car la redondance peut mener à des phases de recherche très longues.

On peut par ailleurs se demander si la valeur d'initialisation des classes n'a pas une influence sur la convergence. En effet, après l'utilisation de la première classe, sa valeur se trouve être inférieure à celle des autres ce qui favorise l'utilisation d'une autre classe et amorce ainsi le processus de convergence. Si après l'application de T_0^1 , on applique à nouveau T_0^1 (on obtient donc la configuration $Cf'_2 = \{2,0,0,0,0,0\}$), la meilleure constellation (Cst) est telle que $M = 0$ pour le premier représentant de Cl_1 et $M = 60$ pour le deuxième. Les deux valeurs se retrouvent dans la bordure de la classe ce qui favorise pour la troisième transition l'application de T_2 . On ne décale donc la convergence que d'une itération (au niveau métaheuristique).

Résultats numériques

La trame de progression, quasi-idéale, décrite précédemment est en fait très peu éloignée de la réalité. En moyenne, le comportement de l'algorithme est très proche de celui présenté et seuls quelques écarts conduisent parfois à explorer d'autres configurations sans pour autant affecter la convergence. Le tableau 15.1 présente un certain nombre de résultats en fonction de différentes valeurs de paramètres :

- le choix du successeur (2ème colonne du tableau 15.1) peut être réalisé par un tirage aléatoire tenant compte de la valeur V_{cf}^2 des configurations (tirage pondéré), ou alors uniforme sur les meilleures configurations voisines (tirage uniforme) (chapitre 13, section 13.8.1).
- la valeur initiale des classes $V_{cl} = 0.5, 0.3, 0.1$

Le caractère aléatoire de certains choix nous conduit à relaiser plusieurs tests pour chacune des valeurs précédentes. Le tableau 15.1 représente la moyenne des résultats obtenus après 15 tests. Les temps de calcul ne sont pas indiqués car ils ne présentent pas d'intérêt : le temps de calcul est engendré par la recherche de bas niveau qui est ici réduite, on obtient des temps de l'ordre de quelques secondes. On constate que le nombre de configurations testées est de l'ordre de 0.3 à 0.7%

Résolution	tirage	V_{cl} initiale	moyenne de $Card(\Delta)$	moyenne de $f(\text{Sol})$
R_1	pondéré	0.5	27	1
R_2	uniforme	0.5	37	1
R_3	pondéré	0.3	35	1
R_4	uniforme	0.3	41	1
R_5	pondéré	0.1	44	1
R_6	uniforme	0.1	51	1

TAB. 15.1 – Résultats de la couche métaheuristique

ce qui représente un très bon pourcentage. Mais ce résultat doit être tempéré par le fait que l'on explore peu de configurations ayant 5 satellites dans la mesure où beaucoup de configurations à 4 satellites conduisent rapidement à une constellation de valeur maximale. On remarque également que la solution est systématiquement trouvée.

La figure 15.6 représente la valeur de la meilleure constellation rencontrée au sein des configurations

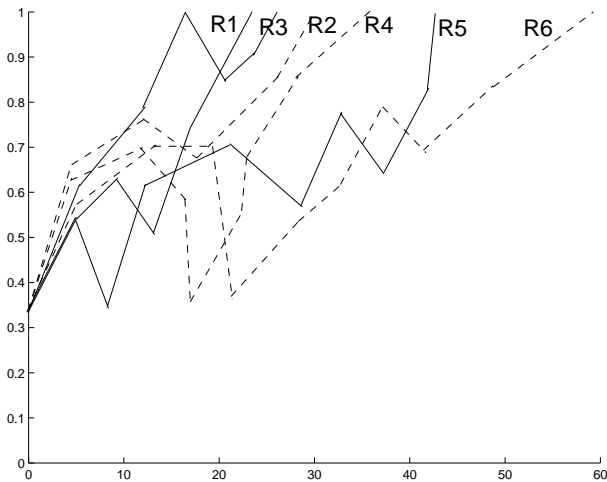


FIG. 15.6 – Valeur de la meilleure constellation

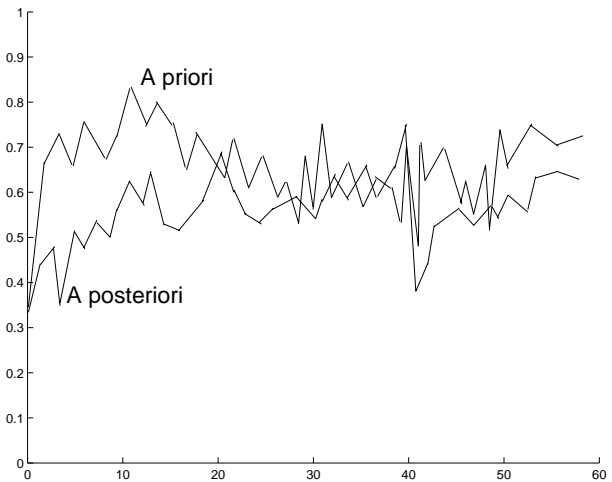


FIG. 15.7 – Valeur à priori et valeur à posteriori

visitées en fonction du nombre d'itérations pour des 6 types de tests réalisés (les valeurs affichées

correspondent à un cas parmi les 15 tests de chaque type réalisés). La figure 15.7 représente la valeur *à priori* et la valeur *à posteriori* des configurations visitées dans le cas R_6 . On constate que des différences apparaissent mais qu'elles s'estompent au fur et à mesure des itérations (on affine la valeur des classes utilisées). De plus, V_{cf}^1 est généralement inférieure à V_{cf}^2 , cela est lié au fait que l'utilisation de deux représentants d'une bonne classe devrait conduire à une bonne configuration (valeur *à priori*) alors que dans notre cas il faut diversifier les classes utilisées.

Il faut par ailleurs noter que pour des raisons de clarté des figures, toutes les valeurs ne sont pas affichées, seules les valeurs correspondant à des changements stratégiques (nature des représentants, nombre de satellites) sont gardées.

15.3.3 Intégration des différents niveaux

Valeur des paramètres

Dans ce cas de figure, nous allons tester l'algorithme global en essayant de répondre au mieux au problème posé : trouver le minimum de satellites géostationnaires pour couvrir la bande de latitude $[-65,65]$. Nous allons donc cumuler les deux résolutions (niveau bas et niveau haut) en fixant les paramètres de chacun des niveaux de la même manière que pour les deux tests réalisés précédemment.

Certains paramètres n'étaient pas fixés au niveau numérique, nous allons leur attribuer une valeur pour effectuer les tests. le seuil (S_{SD}) sur la détection des optimums locaux sera pris égal à 0 (on cherche l'optimum local de manière précise). La durée de la recherche de bas niveau (N_{SD}) pourra prendre plusieurs valeurs que nous donnerons au niveau des résultats numériques. Au cours d'une phase des oscillations stratégiques, nous devons progressivement réduire l'exploration et augmenter l'intensification. Au niveau du test réalisé ici, cela se traduit par une augmentation de N_{SD} à chaque itération des oscillations (on ajoute 2 à sa valeur). En fin de phase, on réinitialise la valeur de N_{SD} . La différence par rapport au cas précédent est que la visite d'une configuration ne renvoie pas nécessairement la meilleure constellation accessible via celle-ci. Autrement dit le retour vers des configurations déjà visitées peut s'avérer intéressant. Pour respecter cette remarque, nous allons fixer la taille k et k' des listes LT_{cf}^1 et LT_{st} respectivement à 2 et 1 : on s'interdit donc de parcourir à nouveau les deux dernières configurations visitées et de supprimer le dernier satellite ajouté.

Résultats

Le tableau 15.2 présente les résultats de plusieurs tests. Les différences entre les cas testés proviennent de la durée initiale de la recherche de bas niveau et de la méthode de sélection du successeur dans le voisinage de la configuration courante. Avant d'analyser ces résultats, il faut préciser que les configurations de l'ensemble Δ sont les configurations visitées au cours de la recherche et qu'une

Résolution	tirage	N_{SD}	moyenne de $Card(\Delta)$	moyenne de $f(Sol)$	temps de résolution
R_1	pondéré	10	281	1	5mn10s
R_2	uniforme	10	215	1	3mn47s
R_3	pondéré	20	127	1	4mn35s
R_4	uniforme	20	113	1	4mn5s
R_5	pondéré	40	97	1	7mn50s
R_6	uniforme	40	80	1	7mn3s

TAB. 15.2 – Résultats de l’algorithme général

même configuration peut être stockée plusieurs fois si la recherche a conduit à revisiter celle-ci. On constate que la solution est systématiquement trouvée ($f(Sol) = 1$). Le nombre de configurations testées est de l’ordre de 0.9 à 3% ce qui est acceptable (surtout si on tient compte de la redondance au sein de Δ). Les différences dans les résultats du tableau 15.2 sont liées au compromis entre la recherche de bas niveau et la recherche de haut niveau : lorsque la durée de la recherche de bas niveau est trop faible (N_{SD}), il n’y a pas convergence vers la constellation optimale au sein d’une configuration à la première visite de celle-ci. Ceci peut conduire à des conclusions partielles erronées sur l’aptitude d’une configuration, à résoudre le problème. Ces constatations sont surtout vraies pour des configurations contenant 2 ou 3 satellites. Lorsque l’on aborde des configurations à 4 représentants, la forte redondance des satellites conduit à une convergence rapide vers une solution de valeur maximale.

Lorsque la durée de la recherche de bas niveau augmente, on augmente également le temps de résolution mais on diminue le nombre de configurations explorées (tableau 15.2). D’après les résultats le meilleur jeu de paramètres semble être de réaliser un tirage uniforme avec une recherche de bas niveau de faible durée (cas R_2) mais il faut noter qu’une augmentation de cette durée permet de réduire de moitié le nombre de configurations visitées sans trop augmenter le temps de résolution (cas R_4) .

15.4 Premières conclusions

Ces premiers résultats indiquent un fonctionnement cohérent de l’algorithme. Mais la faible complexité du problème traité ne permet pas de conclure sur l’efficacité de la recherche dans le cadre de problèmes plus complexes que nous allons traiter maintenant.

Les tests que nous allons réaliser dans la suite du document vont tenir compte de ces premiers résultats et de la démarche justifiée pas à pas jusqu’alors. Dans la suite, nous présenterons uniquement des tests avec des jeux de paramètres correctement choisis. Dans ce cadre, nous soulignons

l'importance du compromis entre les deux algorithmes de recherche. Dans la suite nous retiendrons prioritairement un compromis un peu plus homogène entre la recherche de bas niveau et la recherche de haut niveau (cas R_4 du tableau 15.2) que celui minimisant le temps de résolution (cas R_2). En effet, si ces valeurs de paramètres fournissent un bon résultat dans ce cas d'application, l'augmentation du nombre de classes et de la taille des configurations ferait augmenter de manière importante le nombre de configurations visitées.

Il faut enfin noter que si nous avions eu à concevoir une méthode dédiée à ce problème, l'algorithme aurait été tout autre.

Chapitre 16

Applications en navigation

16.1 Introduction

Au cours de ce chapitre, nous allons adopter la même stratégie d'évaluation que dans le chapitre précédent en réalisant des tests de complexité différente et en évaluant séparément les deux niveaux de l'algorithme avant de réaliser un test général. Le domaine d'application est cette fois-ci la navigation, les problèmes sont donc plus complexes (plus de satellites nécessaires, critère plus long à calculer, interaction forte entre les satellites). Pour juger de la qualité des solutions obtenues, nous avons besoin de comparer celles-ci avec des constellations connues. Contrairement à la première application en télécommunications, aucune solution optimale n'est connue pour les problèmes que nous allons traiter. Nous avons donc décidé de systématiquement comparer la constellation solution à la constellation *GPS* (référence actuelle dans le domaine de la navigation) ou aux résultats d'autres études ([36]).

Les tests présentés au cours de ce chapitre sont les suivants :

1. Ajout de 6 satellites à la constellation de Walker 18/6/1.
2. Ajout de 6 satellites à la constellation GPS.
3. Ajout de n satellites à la constellation de Walker 18/6/1.

16.2 Ajout de 6 satellites à la constellation de Walker 18/6/1

Au cours de ce test nous allons ajouter 6 satellites à la constellation de Walker 18/6/1. L'altitude et l'inclinaison de l'ensemble des satellites manipulés sont celles de la constellation *GPS*. Nous avons choisis cette constellation de Walker car historiquement la constellation *GPS* a été créée de cette manière ([36]). Pour ce premier test, le nombre de satellites ajoutés est fixé. Nous n'explorerons donc que des configurations $Cf = \{Ns_1, \dots, Ns_m\}$ telles que $\sum_{i=1}^m Ns_i = 6$.

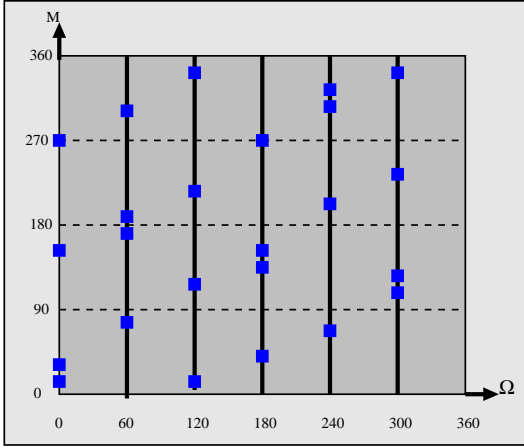


FIG. 16.1 – Constellation Gps

La constellation de Walker est composée de 6 plans orbitaux contenant chacun 3 satellites. La constellation *GPS* est elle aussi composée de 6 plans orbitaux mais comportant chacun 4 satellites. Si pour la constellation de Walker les satellites d'un même plan orbital sont répartis uniformément ce n'est pas le cas pour la constellation *GPS* dont la répartition est donnée figure 16.1. Il se forme un doublet de satellites par plan orbital.

16.2.1 Test du niveau numérique

Valeur des paramètres

Nous considérons dans ce cadre que l'*ODB* est composée de 6 classes. Chaque classe regroupe l'ensemble des positions (M) possibles pour un satellite sur un des 6 plans orbitaux de la constellation de Walker 18/6/1. Pour la classe Cl_i ($i = 1, \dots, 6$) on a donc :

- $a_{min} = a_{max} = 20183km + Re,$
- $e_{min} = e_{max} = 0,$
- $i_{min} = i_{max} = 55,$
- $\omega_{min} = \omega_{max} = 0,$
- $\Omega_{min} = \Omega_{max} = \frac{2\Pi}{6}(i - 1),$
- $M_{min} = 0, M_{max} = 360.$

Pour tester le niveau numérique de l'algorithme nous nous plaçons dans la configuration $Cf = \{1,1,1,1,1,1\}$. C'est cette configuration qui permet de s'approcher (structurellement) le plus de la constellation *GPS* (étant donné que les satellites de la constellation de Walker sont fixés et que leurs positions ne coïncident pas avec celles de satellites de la constellation *GPS*, il est impossible de retrouver cette dernière à partir de la configuration Cf_1).

Nous avons donc "uniquement" à déterminer la position des 6 satellites sur l'orbite (M) : nous cherchons la manière optimale d'ajouter 1 satellite par plan à la constellation de Walker 18/6/1.

La constellation sera optimisée, de manière séparée, selon deux critères (chapitre 5) : le *PDop* ([29]), représentant la précision de positionnement, et la disponibilité, qui intègre la notion de pannes. On cherche à minimiser le *PDop* et à maximiser la disponibilité. Par ailleurs, la disponibilité sera

calculée en tenant compte de 6 cas de panne tirés au hasard, la performance minimale attendue est par ailleurs $PDOP < 6$.

La fonction d'évaluation d'une constellation de navigation est généralement binaire et valide par exemple la constellation si le calcul de $PDop$ en chaque point d'intérêt ne dépasse pas une certaine valeur, dans le cas contraire la constellation est rejetée. Dans notre cas, pour les raisons de régularité invoquées dans le chapitre 12, nous allons utiliser une fonction d'évaluation d'une constellation (f) qui renvoie la moyenne du $PDop$ au cours de la simulation (nous verrons dans la section 16.2.2 comment analyser plus finement cette simulation).

La configuration $Cf = \{1,1,1,1,1,1\}$ comporte beaucoup d'optimums locaux, la recherche de bas niveau va donc différer par rapport à celle utilisée dans le premier exemple du chapitre précédent. Nous allons fixer un seuil S_{SD} dont la valeur n'est pas nulle ($S_{SD} = 0.1$), ce qui nous permettra de stopper la descente de plus forte pente, dès que la solution ne progressera plus assez, pour repartir vers une autre solution.

Résultats

Les paramètres du processus de simulation sont fixés de manière à ce que l'évaluation d'une constellation avec un critère de type $PDop$ soit de l'ordre d'une seconde sur le calculateur utilisé. Les paramètres ainsi fixés ne permettent pas une évaluation assez fine de la constellation pour qu'elle puisse être validée complètement (critère d'acceptation de type pire cas sur la simulation). Mais étant donné que nous réalisons un calcul de moyenne du critère pour apprécier globalement les performances des constellations visitées, et que le temps de simulation se trouve être considérablement réduit par rapport à une évaluation fine nous nous contenterons de ces valeurs. Le calcul de la disponibilité prend plus de temps que celui du $PDop$ mais des calculs adroits et des restrictions sur le nombre de pannes permettent de limiter ces calculs (de l'ordre de quelques secondes).

Par ailleurs, étant donné que la solution optimale n'est pas connue (en supposant qu'il n'y en ait qu'une seule), nous avons décidé de stopper la recherche après un certain nombre de constellations visitées (1000) et de comparer la meilleure solution obtenue (notée Sol) pour différents jeux de paramètres. Le tableau 16.1 présente les résultats pour 6 jeux de paramètres. Pour chaque cas 15 tests ont été réalisés et le tableau présente une valeur moyenne des résultats obtenus. On rappelle que pour un critère de type $PDop$, on calcule la moyenne du $PDop$ au cours de la simulation et que pour un critère du type $Dispo$, on calcule le pourcentage de temps pendant lequel la constellation fournit les performances attendues. Contrairement aux résultats obtenus dans le cadre du premier problème de télécommunications traité, lorsqu'à une itération on modifie un paramètre dans un direction donnée, on peut être amené à le modifier dans la direction opposée quelques itérations après. Par conséquent l'algorithme de bas niveau nécessite de nombreux changements de directions avant de converger vers un optimum local. Les valeurs basses du pas de variation de M lors de

Résolution	Critère	Pas de variation	$f(Sol)$
R_1	<i>PDop</i>	5ř	5.9568
R_2	<i>Dispo</i>	5ř	99.2358
R_3	<i>PDop</i>	10ř	5.7265
R_4	<i>Dispo</i>	10ř	99.7859
R_5	<i>PDop</i>	20ř	5.6533
R_6	<i>Dispo</i>	20ř	99.5496

TAB. 16.1 – Résultats de la couche numérique

la descente de plus forte pente (5 et 10 degrés), permettent d'avoir une valeur précise de M mais requièrent un nombre d'itérations considérable. Dans de telles conditions, après 1000 itérations les meilleures solutions atteintes (cas R_1 et R_2) ne sont pas de très bonne qualité en comparaison avec les autres résultats. Par contre, un pas de 20 (cas R_5 et R_6) permet de converger plus rapidement vers un optimum local et ainsi repartir vers d'autres constellations. Dans notre cas cela permet d'atteindre une meilleure solution pour un même nombre d'itérations.

On peut donc conclure dans ce cas précis qu'un calcul fin des paramètres orbitaux n'est pas performant alors que des variations de forte amplitude (20) permettent de détecter plus rapidement des régions d'intérêt au sein de la configuration Cf_1 . Si on compare ces résultats à la constellation *GPS*, on obtient une valeur moyenne de *PDOP* de 5.1542 et une disponibilité de 99.8554%. Le résultat concernant la disponibilité est supérieur à ceux que nous avons obtenus mais il faut rappeler que les solutions sont nécessairement différentes (par construction). Par ailleurs le résultat sur le calcul de *PDop* est comparable à celui fournit par la constellation *GPS*.

Nous allons maintenant analyser les résultats en terme de structure de constellation. Les figures 16.2 et 16.3 présentent les constellations obtenues dans deux cas particuliers des tests R_5 et R_6 (nous avons retenu les cas particuliers les plus intéressants). On constate que dans le cas du critère de type *PDop* (figure 16.2) la constellation obtenue n'a pas la structure de la constellation *GPS* mais tend à uniformiser la constellation. Par contre, dans le cas du critère de disponibilité (figure 16.3) la solution fait apparaître des doublets de satellites. Ce résultat est intéressant dans la mesure où il tend à valider la constellation *GPS* en proposant une structure de même nature. La formation de doublets permet de pallier la défaillance d'un satellite.

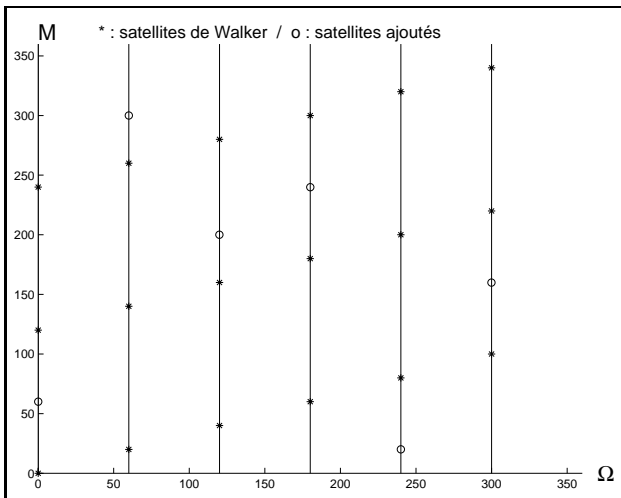


FIG. 16.2 – Critère : $PDop$, pas : 20

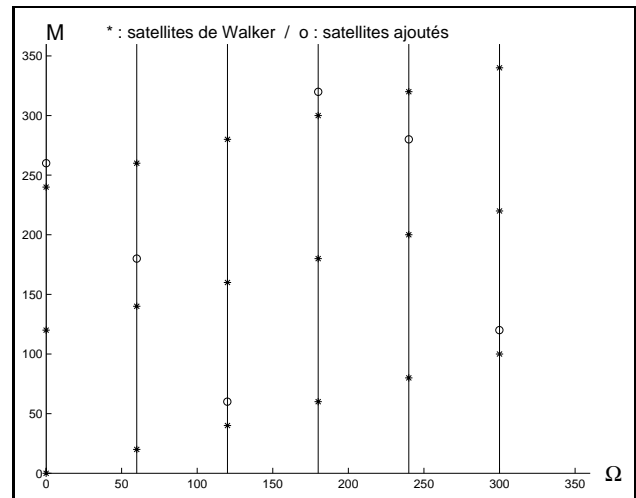


FIG. 16.3 – Critère : $Disponibilité$, pas : 20

16.2.2 Test du niveau métaheuristique

Valeur des paramètres

Les paramètres utilisés pour tester ce niveau sont définis de la manière suivante :

- L'*ODB* utilisée est composée des 6 classes du test précédent.
- A chaque satellite nous associons la valeur V_{s3} (V_{s4}) (section 13.5.2 chapitre 13) qui représente le nombre de calculs de précision, pour lesquels le satellite considéré a participé, et tels que $PDOP > 6$ (≤ 6). Ces deux valeurs permettent de différencier les satellites au sein de la constellation afin de sélectionner ceux qui doivent être supprimés ou modifiés.

Comme indiqué en début de section, nous n'explorons que des configurations à 6 satellites (que l'on ajoute aux 18 de la constellation de Walker). Cela revient à fixer $n_{min} = n_{max} = 6$, ce qui interdit l'utilisation des transitions de type T_0 et T_1 . C'est donc un algorithme de haut niveau spécifique que nous allons tester ici. La taille de l'espace de recherche n'en reste pas moins importante puisqu'il y a $6^6 = 46656$ configurations possibles.

Comme pour les tests du chapitre précédent, nous devons inhiber l'influence de la recherche de bas niveau en assurant que celle-ci renvoie la meilleure constellation accessible via la configuration explorée. La présence de nombreux optimums locaux au sein d'une configuration et le manque d'informations sur la nature de l'optimum global nous conduisent à brider les classes utilisées en autorisant que quelques valeurs pour l'anomalie M des satellites. Nous allons choisir ces valeurs en fonction de la structure de la constellation *GPS* en ne conservant que les positions conduisant à des doublets. Il peut se former trois doublets par plan (un doublet par satellite déjà en place), nous ne garderons donc que trois positions par plan orbital. Par ailleurs, l'écart angulaire avec le satellite

formant le doublet est fixé à 10.

Pour chaque configuration visitée il y a donc $3^6 = 729$ constellations possibles (3 positions pour chaque satellite ajouté). Parmi ces 729 constellations certaines sont identiques à cause des symétries de la constellation de Walker. Pour inhiber l'influence du niveau bas nous devons évaluer toutes les constellations différentes pour retourner la meilleure. L'évaluation d'une configuration sera donc un processus long mais exact.

Enfin, tenant compte des enseignements du chapitre précédent (section 15.4), nous allons utiliser un tirage aléatoire uniforme pour sélectionner le successeur de la configuration courante et une durée d'oscillation stratégique (N_{os}) de 50 itérations.

Résultats

D'après les données précédentes, il faut plusieurs minutes pour évaluer (de manière exhaustive) une configuration. Le temps étant un paramètre linéaire du nombre de configurations visitées, nous ne parlerons que de ce dernier. Le tableau 16.2 présente deux résultats différents (par critère) représentant des structures différentes obtenues après application de l'algorithme de haut niveau. L'arrêt de la recherche est déterminé par l'utilisateur. La meilleure solution obtenue est alors présentée ainsi que le nombre d'itérations parcourues avant son apparition.

Cas testé	$Card(\Delta)$	$f(Sol)$
<i>PDop1</i>	1058	5.8547
<i>PDop2</i>	1376	5.5635
<i>Dispo1</i>	903	99.7926
<i>Dispo2</i>	1039	99.8253

TAB. 16.2 – *Résultat niveau métaheuristiques*

Nous constatons que pour le critère de disponibilité les valeurs des solutions (lignes 3 et 4) sont comparables à celle du meilleur scénario des tests précédents. Par contre ce n'est pas tout à fait le cas pour le *PDOP* pour lequel les valeurs sont moins bonnes, cela est du à la structure forcée de la constellation (formation de doublets).

Concernant les structures, on note deux types de constellations : des structures dont les doublets sont répartis sur chacune des classes (avec des différences de position par rapport à la constellation *GPS*) (figure 16.4) et des structures où les doublets sont localisés sur 2 ou 3 plans (figure 16.5). Si la performance, relative, du premier type de constellation est justifiée par la structure de la constellation *GPS*, le second type de constellation demande une analyse plus poussée. Si on analyse ce dernier type de résultat avec des calculs de type pire cas de *PDop* sur la simulation, on se rend compte que les performances chutent. La relativement bonne évaluation de ce type de constellation

est due à la présence de 2 ou 3 plans qui contribuent de manière très active aux bonnes performances de la constellation (beaucoup de satellites sur ces plans), par contre les autres plans engendrent des résultats de moins bonne qualité. Le critère d'optimisation utilisé, réalisant une moyenne des performances au cours du temps, ne permet pas d'écarter ce type de constellation. On ne peut ici empêcher l'apparition de telles constellations sans modifier la manière de les évaluer. On note tout de même que le résultat obtenu est cohérent avec le critère d'optimisation.

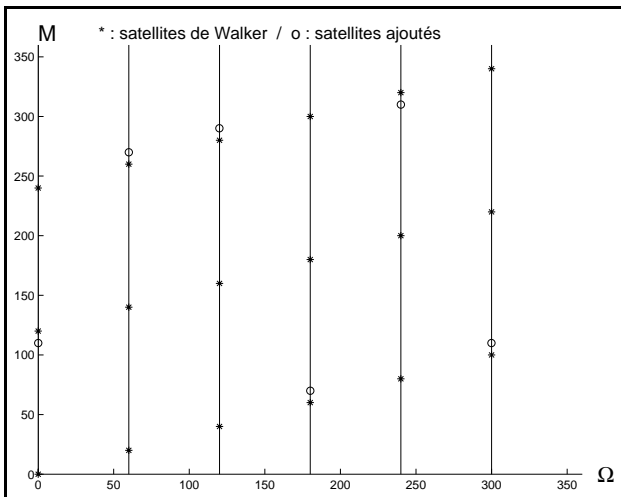


FIG. 16.4 – *Premier type de structure*

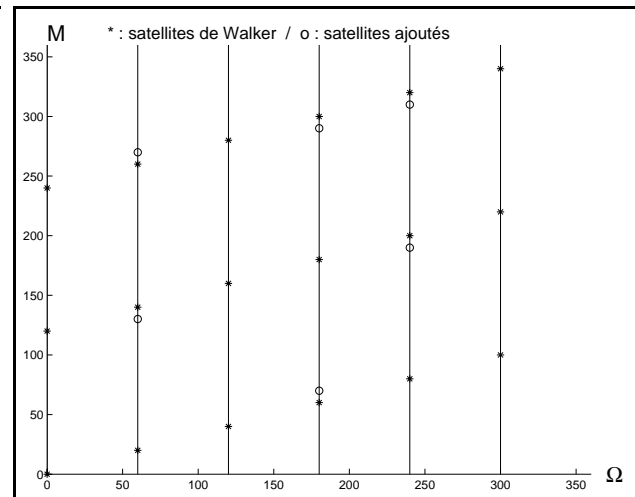


FIG. 16.5 – *Deuxième type de structure*

16.2.3 Intégration des différents niveaux

Valeur des paramètres

Nous allons maintenant intégrer les différents niveaux. Nous conservons les 6 classes décrites dans le test du niveau numérique mais sans les brider comme dans le test du niveau métaheuristique. Les temps de calculs associés à l'évaluation d'une configuration sont réduits par rapport aux tests du niveau métaheuristique dans la mesure où on applique l'algorithme de bas niveau durant N_{SD} itérations (on n'évalue donc pas 729 constellations par configuration). La valeur de N_{SD} est fixée à 50 au début d'une phase des oscillations stratégiques pour atteindre 200 en fin de phase. Cette phase a une durée identique au cas précédent ($N_{os} = 50$). Durant les itérations des oscillations stratégiques, on réduit l'espace de recherche, le nombre de configurations visitées diminue donc. Après une phase des oscillations stratégiques nous avons parcourus beaucoup plus de configurations avec 50 itérations de l'algorithme de bas niveau que de configurations avec 200 itérations ce qui contribue également à réduire le temps total de résolution.

Par ailleurs, tenant compte des remarques de la section 16.2.1 nous adoptons un pas de recherche

de bas niveau de 20.

Résultats

Nous avons fixé la valeur des paramètres afin de respecter un compromis entre la recherche de bas niveau et la recherche de haut niveau. Sous ces conditions, l'exploration d'une configuration n'est pas réalisée en profondeur à chaque visite et ne conduit donc généralement pas à la meilleure constellation. Pour évaluer les résultats de l'algorithme nous avons décidé de stopper la recherche après avoir exploré 3% des configurations.

Les deux premières lignes du tableau 16.3 présentent les résultats ainsi obtenus. Les résultats sont

Cas testé	$Card(\Delta)$	$f(Sol)$
<i>PDop1</i>	1400	5.4587
<i>Dispo1</i>	1400	98.3697
<i>PDop2</i>	1246	5.1855
<i>Dispo2</i>	1386	98.5213

TAB. 16.3 – *Résultat de l'algorithme général*

légèrement moins bons que pour les tests de l'algorithme métaheuristique (performances moins bonnes, plus d'itérations). Après analyse il s'avère que la meilleure constellation rencontrée l'ait été après qu'une configuration déjà visitée le soit à nouveau. On peut donc penser que la recherche de bas niveau n'était pas assez poussée lors de la première visite. Nous avons discuté du compromis entre les deux niveaux de recherche et il est apparaît évident qu'une augmentation de la durée de recherche de bas niveau permettrait de trouver une solution de meilleure qualité pour chaque configuration explorée. Mais le prix à payer serait une augmentation inacceptable du temps de calcul.

Étant donné que les configurations visitées plusieurs fois sont des configurations dont l'évaluation à *posteriori* (V_{cf}^1) est de bonne qualité, nous pourrions systématiser ce retour vers les meilleures d'entre elles, à la fin d'une phase des oscillations stratégiques par exemple, afin de pousser l'exploration. Nous avons réalisé un certain nombre de tests en appliquant ce nouveau principe et des résultats sont donnés dans les deux dernières lignes du tableau 16.3. Les résultats présentés sont meilleurs que les précédents en terme de nombre de configurations testées et de valeur de solution. Mais cela ne doit pas laisser penser que l'amélioration est systématique, et le résultat doit être tempéré par l'existence de retours infructueux qui conduisent à une augmentation du temps de résolution.

Enfin au niveau structurel, les solutions obtenues sont proches de celles obtenues précédemment. Il faut noter que les temps de résolution sont très longs (environ deux jours) à cause des temps de simulation des critères.

16.3 Ajout de 6 satellites à la constellation *GPS*

Nous allons maintenant appliquer notre algorithme à la résolution du problème suivant : ajouter 6 satellites à la constellation *GPS*. Nous comparerons les résultats obtenus avec ceux de l'étude réalisée par *Alcatel* et portant sur le même problème ([36], [78]). Les hypothèses faites pour la résolution du problème sont synthétisées dans le tableau 16.4 :

Hypothèse	Études précédentes	Notre Approche
Méthode de résolution	algorithmes génétiques	algorithme 13.2
Altitude des satellites	altitude <i>GPS</i>	altitude <i>GPS</i>
Inclinaison des satellites	inclinaison <i>GPS</i>	inclinaison <i>GPS</i>
excentricité et argument du périégée	0	0
Position des satellites ajoutés	1 par plan	appartenant à une classe de l' <i>ODB</i>

TAB. 16.4 – *Ajout de 6 satellites à la constellations GPS : hypothèses*

16.3.1 Résultats

La complexité du problème est identique au cas précédent puisque l'on ajoute 6 satellites et que l'*ODB* comporte 6 classes. On constate dans les résultats, que deux types de solutions émergent de notre algorithme. Dans les deux cas les constellations solutions sont majoritairement des représentants de la configuration $\{1,1,1,1,1,1\}$. Dans le premier cas (figure 16.6), les satellites ajoutés ne forment pas de doublets avec les satellites de la constellation *GPS* déjà en place. Ils tendent à se positionner à l'opposé du doublet déjà en place, sur le plan considéré, et à égale distance des deux satellites adjacents. Dans le second cas (figure 16.7), c'est une structure plus proche de *GPS* qui apparaît avec formation de deux doublets par plan orbital. Cette dernière solution est également proche de celle obtenue par l'étude [36]. Le tableau 16.5 donne la valeur de la disponibilité obtenue. Les résultats sont un peu moins bons que dans le cas de l'étude [36] mais restent de très bonne qualité surtout si on rapporte ceux-ci aux hypothèses formulées précédemment. La classe d'appartenance des satellites n'étant pas connue. On présente ici des structures obtenues de façon majoritaire

Cas testé	Disponibilité
Études précédentes	99.924
Notre Approche cas 1 (43%)	99.8995
Notre Approche cas 2 (31%)	99.9145

TAB. 16.5 – *Résultats*

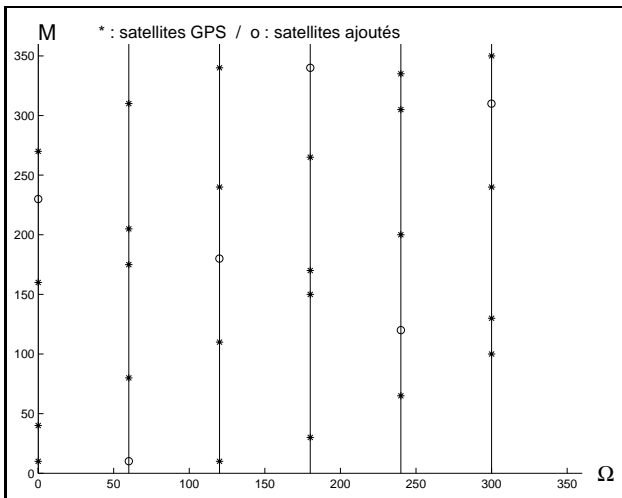


FIG. 16.6 – *Notre approche : cas 1*

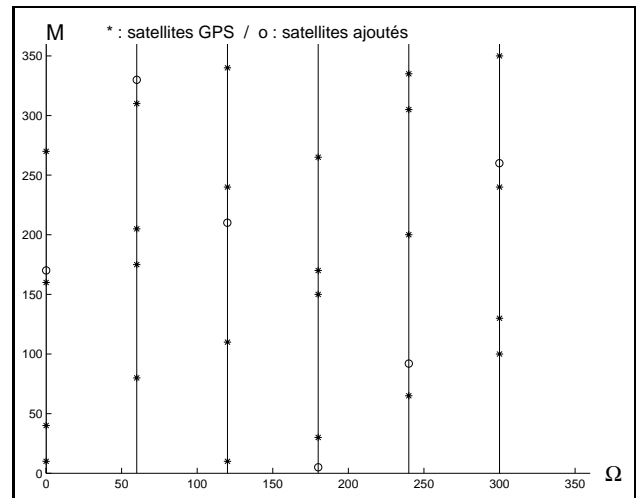


FIG. 16.7 – *Notre approche : cas 2*

(74% des cas) mais d'autres solutions peuvent être proposées (26% des cas). L'évaluation est alors sensiblement plus faible et la répartition des satellites plus désordonnée. On constate dans ce cadre que la structure des solutions ne se conserve pas d'une phase des oscillations stratégiques à la suivante (manque de progression des solutions et donc diversification). Lors de l'obtention des deux solutions des figures 16.6 et 16.6 la structure se conserve durant plusieurs phases ce qui permet une convergence vers les solutions proposées.

Les temps de résolution sont comparables au cas précédent (environ 2 jours) et le nombre de configurations visitées également (inférieur à 1500).

16.4 Extension au cas de n satellites

16.4.1 Principe

On généralise maintenant le premier cas testé en considérant l'ajout de n satellites ($n \in [1,12]$) au lieu de 6. On test donc ici l'algorithme sur un problème plus complexe que ceux résolus jusqu'alors dans la littérature, puisque le nombre de satellites est inconnu, et ce avec le même algorithme. Les autres paramètres restent identiques à ceux de la section 16.2 et notamment la composition de l'*ODB* (6 classes). Le critère utilisé ici est le critère de disponibilité.

La taille de l'espace de recherche du niveau heuristique se trouve considérablement augmentée : on est passé de la résolution du problème T_6^6 , de cardinalité 46656, à celle du problème $P_6^{1,12}$, de cardinalité 1865773440 (cf chapitre 13). Nous comparerons les résultats à ceux obtenus lors des deux tests précédents (ajout de 6 satellites à la constellation de Walker 18/6/1 et ajout de 6 satellites

à la constellation *GPS*). Nous présenterons uniquement le test sur l'algorithme global. Les tests séparés des différents niveaux ne présentent aucune nouveauté par rapport au premier cas.

16.4.2 Résultats

Lors des tests réalisés nous avons donné la priorité aux performances de la constellation : on cherche à maximiser la disponibilité de la constellation. Afin de forcer l'obtention de constellations dont la disponibilité est proche du maximum, nous allons considérer qu'une solution est viable si sa disponibilité est supérieure à un certain seuil (99.7%).

Étant donnée la taille de l'espace des configurations, nous souhaitons que l'algorithme explore un taux très faible de celles-ci. Nous allons donc contraindre la recherche en paramétrant l'algorithme de la manière suivante :

- pour éviter de revenir sur des configurations déjà visitées, nous allons fixer la taille de la liste LT_{Cf}^1 à 100 (liste des configurations récemment visitées),
- afin d'assurer que la première visite soit représentative nous allons fixer la durée de la recherche de bas niveau entre 100 et 150 itérations. De plus les pas de variation sont choisis de manière à rapidement échantillonner les configurations plutôt que de fournir des valeurs précises des paramètres,
- pour écarter des régions de l'espace nous allons définitivement positionner les configurations de la liste LT_{Cf}^2 à l'état tabou (liste des configurations de mauvaise qualité),
- nous décréterons qu'une configuration est de mauvaise qualité dès-lors que sa valeur (qui n'est autre que la valeur moyenne des constellations visitées appartenant à celle-ci) sera inférieure à 99.7.

Ces considérations particulières permettent à l'algorithme de haut niveau, et particulièrement aux oscillations stratégiques, de rapidement déterminer le nombre de représentants nécessaires (figure 16.9) : l'ajout d'un satellite augmente systématiquement la valeur du critère, puisqu'ajouter un satellite permet de lutter contre les cas de pannes. On obtient ainsi rapidement des configurations dont les représentants sont viables (figure 16.8). La réduction de l'intervalle de variation $[n_{min}, n_{max}]$ conduit à écarter des ensembles de configurations notamment à travers la constitution de la liste LT_{Cf}^2 . Les oscillations stratégiques ainsi paramétrées (mise à l'état tabou définitive, ...) permettent de guider de manière plus directe la recherche et conduisent à l'obtention de configurations à 12 représentants (figure 16.8). Le temps d'évaluation d'une constellation est de l'ordre de 15 secondes (tirage de 12 cas de pannes) ce qui génère des temps de résolution de plusieurs jours (4 à 6 jours). Les solutions sont obtenues après une moyenne de 18 phases d'oscillations stratégiques comportant chacune 10 configurations visitées. Le nombre de satellites nécessaires (12) est en moyenne obtenu après 2 phases d'oscillation stratégiques (figures 16.8 et 16.9). Les 16 phases restantes permettent de faire varier la structure des configurations visitées en interdisant ou privilégiant un certain nombre

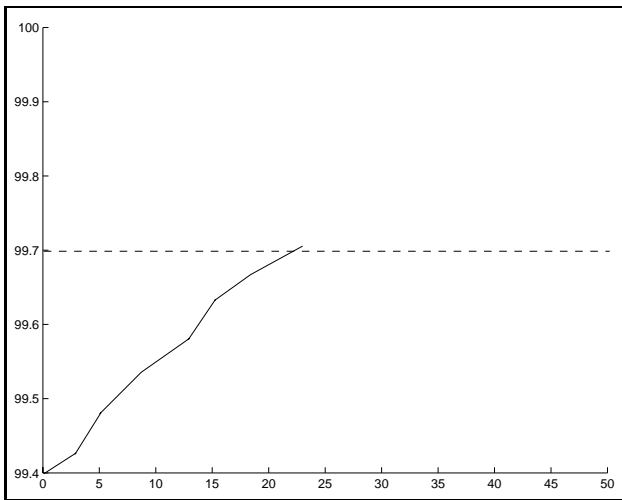


FIG. 16.8 – Valeur à posteriori des configurations

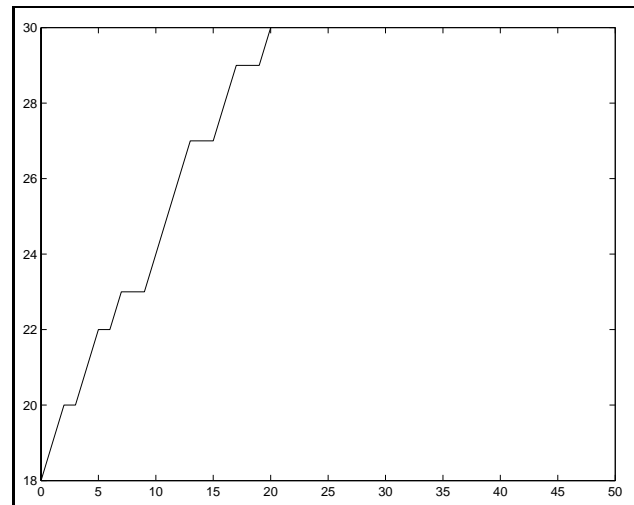


FIG. 16.9 – Nombre de satellites

de classes. Les meilleures constellations ainsi obtenues permettent de rivaliser avec les résultats précédents puisque l'on obtient une disponibilité de 99.9265 (contre 99.924 pour l'étude précédente).

16.5 Extension du nombre de classes

16.5.1 Principe

Nous allons compliquer un peu plus le problème en doublant le nombre de classes de l'*ODB*. On autorise maintenant, l'ajout de satellites sur des plans différents de ceux des constellations de Walker et *GPS*. On permet ainsi de s'éloigner un peu plus de la structure de la constellation *GPS*. Les 6 classes ajoutées correspondent aux plans intermédiaires entre deux plans de la constellation de Walker 18/6/1 : $\Omega_{min} = \Omega_{max} = \frac{2\pi}{12}(2i + 1)$ pour $i = 0, 1, 2, 3, 4, 5$.

Par ailleurs, les hypothèses faites sont les suivantes

- ajout de 6 à 12 satellites,
- la constellation initiale est la Walker 18/6/1,
- les oscillations stratégiques sont paramétrées de la même manière que dans la section 16.4.

16.5.2 Test du niveau numérique

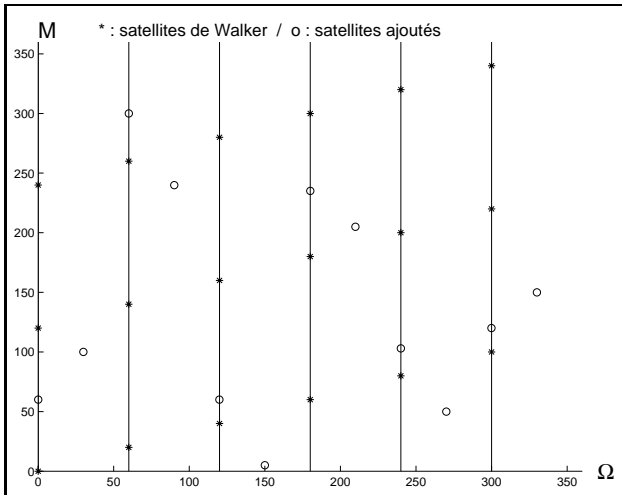


FIG. 16.10 – Niveau numérique

Nous avons décidé de tester une configuration particulière

($Cf = \{1,1,1,1,1,1,1,1,1,1,1\}$) dans le cadre de cette étude. Cette configuration comporte un représentant de chaque classe et la structure majoritairement obtenue (figure 16.10) fait apparaître des clusters de satellites. C'est à dire que les doublets formés sur un plan orbital sont complétés par le satellite du plan adjacent. La disponibilité maximale obtenue dans ce cas est de 99.8947. Cette valeur est inférieure à la précédente mais la finesse des calculs de bas niveau ne permet pas d'atteindre une précision suffisante dans la valeur des paramètres orbitaux.

16.5.3 Test global

Nous traitons ici le problème $P_{12}^{1,12}$ dont la cardinalité est de l'ordre de 10^{13} , contre 10^9 pour le problème précédent. Les paramètres utilisés pour les oscillations stratégiques permettent de rapidement converger vers une configuration à 12 satellites (en 2 phases des oscillations stratégiques).

Une différence apparaît dans la structure des constellations obtenues. Ce n'est en effet pas la formation de deux doublets par plan orbital qui semble être l'optimum le plus attracteur. Bien au contraire, on constate que les satellites ont majoritairement tendance à se placer sur les plans intermédiaires. La figure 16.11 illustre ce ré-

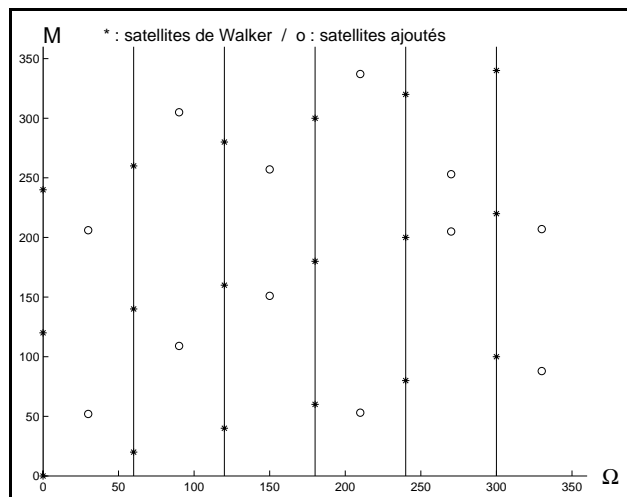


FIG. 16.11 – Résultat global

sultat pour un cas particulier ayant conduit à l'obtention des 12 satellites sur les plans intermédiaires. Ce résultat ne reflète en aucune manière un manque d'efficacité de notre algorithme. La constellation *GPS* est uniquement prise comme base de comparaison et son optimalité vis-à-vis du critère utilisé (la disponibilité) n'est pas assurée. La convergence de notre algorithme vers une constellation structurellement différente tend seulement à prouver qu'il existe d'autres optimum locaux forts. Par ailleurs, la disponibilité associée à cette constellation est de 99.8978%.

16.6 Conclusion

Des résultats obtenus au travers de ces différents tests, nous pouvons donner plusieurs conclusions

- La convergence vers une solution est fortement guidée par le paramétrage de l'algorithme et notamment par le compromis entre la résolution de niveau bas et celle de niveau haut.
- Les temps de calcul restent très contraignants malgré les efforts mis en oeuvre pour les alléger.
- L'algorithme fonctionne de manière cohérente, et intègre les différentes structures que nous avons mis en place (oscillations stratégiques, mémoire, recherche locale, diversification, ...).
- L'algorithme a permis de résoudre un problème de nature plus complexe que celui de l'étude [36].

Perspectives et conclusions

Conclusions sur les résultats

Le premier constat à faire est que les divers paramètres de notre algorithme sont fixés par l'expérience. Celle-ci nous montre (chapitres 15, 16) que le jeu des compromis entre les différents niveaux est très important. L'utilisateur joue donc un rôle important et, tout comme l'algorithme, il doit utiliser son expertise et sa mémoire pour paramétrer (via l'*ODB* et autres paramètres de l'algorithme) au mieux l'algorithme.

Malgré cette expertise nécessaire pour mettre en oeuvre l'algorithme (expertise spatiale pour l'*ODB* et expertise expérimentale pour les paramètres de l'algorithme), celui-ci présente un aspect plus général que les algorithmes que l'on trouve dans la littérature dans le domaine du design de constellation. Tous les tests ont été effectués avec le même algorithme, seul le paramétrage a changé. Un des points forts de l'algorithme est de se passer de la connaissance du nombre de satellites requis pour résoudre le problème : la recherche du nombre de satellites est gérée au niveau des oscillations stratégiques. Cet aspect a permis entre autre de traiter le problème complexe de l'ajout de n satellites à une constellation (n inconnu).

De manière générale, nous avons montré, au cours de ces tests, un fonctionnement cohérent de l'algorithme à travers des tests de nature et de complexité différentes :

- intégration et interaction des différents niveaux,
- obtention de solutions comparables à des solutions connues (lorsqu'on dispose d'une telle connaissance).

Enfin, la modélisation et les éléments de résolution présentés dans les chapitres 11, 12 et 13 ont été implantés sur machine (avec les contraintes que toute mise en oeuvre implique) dans un ensemble de logiciels permettant de définir le problème traité (zones d'intérêt, critères utilisés,...), de paramétrer l'algorithme (simulation, *ODB*, niveau numérique, niveau métaheuristique,...), de réaliser un suivi de son déroulement (état de la mémoire, configurations visitées, progression de la solution, ...) et de visualiser les résultats (visualisation des courbes de valeur, des configurations visitées, des solutions obtenues). Quelques sorties graphiques ainsi que la structure logiciel sont présentées en annexe

(Annexe 2 page 235).

Limitations

Plusieurs types de limitations apparaissent dans notre algorithme. Nous rappelons d'abord l'inadéquation de celui-ci pour traiter des problèmes d'observation. L'incompatibilité vient du critère utilisé, le délai de revisite maximal, qui ne permet pas d'évaluer les satellites indépendamment les uns des autres.

Ensuite, lorsqu'on applique l'algorithme à la résolution d'un problème complexe (nécessitant beaucoup de satellites) en ne prenant pas le soin contraignant de fournir à l'algorithme une constellation initiale, celui-ci se trouve confronté à deux problèmes : déterminer une constellation viable, et optimiser celle-ci. Si nous avons abstrait le premier problème dans le cadre de la navigation (chapitre 16) en partant d'une constellation de Walker, ceci n'est pas toujours possible ou intéressant. On constate lors de l'application de l'algorithme en partant d'une configuration vide que les constellations obtenues ne permettent pas de couvrir tous les points d'intérêts. Ces points non satisfaits sont souvent dispersés sur toute la zone d'intérêt et l'algorithme peine à trouver une orbite permettant de les satisfaire. Nous présenterons dans la prochaine section une perspective visant à résoudre cet obstacle.

Par ailleurs, l'algorithme exprime plus de difficultés à converger lorsque les configurations optimales (ou sous-optimales) font intervenir beaucoup de classes, chacune ayant peu de représentants (configuration du type $\{1,1,1,1\}$). Dans ce cadre, c'est l'association des différentes classes qui assure les bonnes performances de la constellation. Lorsque la configuration optimale met en jeu un sous-ensemble restreint de classes dont le nombre de représentants est important (configuration du type $\{4,0,0,0\}$), l'algorithme converge de manière plus directe. Nous présenterons dans la prochaine section une perspective visant à repousser cette limitation.

Enfin, les temps de calcul sont encore trop importants pour envisager des résolutions plus complexes. Il faut donc continuer l'effort réalisé pour réduire le temps d'évaluation des critères (optimisation du code, simplification des calculs, ...), et utiliser des calculateurs plus puissants (calculateurs parallèles par exemple).

Contributions

Les contributions à caractère innovant se décomposent en contributions pratiques et théoriques :

- la modélisation développée au chapitre 11 est un point fondamental. A notre connaissance, aucune démarche comparable n'était présente dans l'état de l'art. L'*ODB*, fournit en outre un ensemble d'idées et de fonctionnalités réutilisables dans le contexte spatial.

- la méthode de résolution dédiée à la modélisation précédente est également originale. Les aspects avancés de la recherche tabou y sont intégrés et sont à l’origine des meilleures performances observées. Le caractère général de l’algorithme proposé permet de l’appliquer à différents problèmes.
- en termes de constellations obtenues, des solutions connues ont été retrouvées. Par ailleurs, l’algorithme a su fournir des résultats performants dans un cadre plus général (ajout d’un nombre inconnu de satellites). Ce dernier point était un des objectifs initiaux.

Par ailleurs, l’étude de l’existant en matière d’optimisation de constellations de satellites a donné lieu à un certain nombre de contributions de second ordre:

- travail sur les critères (simplification, codage, ...).
- étude et nouvelle implantation de la méthode de Hanson.
- développement d’un ensemble de logiciels permettant le paramétrage, le suivi et la visualisation des résultats de l’algorithme.

Perspectives

Malgré les limitations présentées par la résolution, la modélisation apporte de nouvelles perspectives. Au cours de ce travail doctoral nous avons analysé les différents problèmes posés

- par le domaine d’application (taille de l’espace de recherche, lourdeur des évaluations, ...),
- par les méthodes utilisées (trop d’évaluations, algorithmes naïfs, ...),

et nous avons proposé et mis en oeuvre un élément de réponse pour chacun de ces obstacles

- simplification des critères,
- augmentation du nombre de valeurs retournées après la simulation,
- recherche locale,
- utilisation de structures de mémoire,
- décomposition de l’espace de recherche.

Certains choix concernant les méthodes utilisées pour la résolution ont été guidés par le souci de minimiser les évaluations. Nous sommes donc passés d’une recherche globale (évolution d’une population de solutions) à une recherche locale (évolution d’une solution unique). Mais il faut noter que l’ensemble des avancées réalisées peuvent être répercutées sur d’autres algorithmes, y compris les algorithmes génétiques. Dans le cadre de ces derniers, on peut par exemple envisager d’améliorer l’opérateur de croisement à l’aide de l’analyse de la simulation, des structures de mémoire ou encore de la décomposition de l’espace de recherche (Approche Scatter Search).

Par ailleurs, deux perspectives sont en cours d’étude pour repousser les limites des méthodes de résolution et étendre la modélisation.

Dans un premier temps, l'existence d'un certain nombre de points d'intérêt non satisfaits laisse imaginer un calcul spécifique de l'orbite permettant de les satisfaire. Dans ce cadre, nous envisageons d'utiliser une estimation de paramètres permettant de déterminer les paramètres orbitaux en fonction des points à satisfaire. L'estimation consiste donc à prendre en entrée des points en deux dimensions (longitude et latitude) et à estimer a, e, i, ω, Ω , et M tels que la trace au sol associée permette de couvrir un maximum de points. Les figures 16.12 et 16.13 présentent un cas particulier pour lequel il est possible de trouver une orbite permettant de satisfaire l'ensemble des points, les figures 16.14 et 16.15 présentent un ensemble de points pour lesquels il n'est pas possible de trouver une unique orbite. Deux modèles ont été mis au point, l'un uniquement basé sur les coordonnées longitudinales des points, l'autre intégrant une donnée temporelle (Annexe A). Par ailleurs plusieurs estimateurs ont été utilisés :

- Les estimateurs *OLS* illustrés par les figures 16.12 à 16.15
- Les estimateurs *ODR* qui permettent de pondérer les erreurs sur les entrées. On peut ainsi faire une différence entre les erreurs sur les données temporelles et spatiales.
- Les M-estimateurs permettant une estimation robuste capable de gérer un important volume de données aberrantes en entrée de l'estimateur.

Le lecteur est renvoyé à l'Annexe 1 page 229 pour plus de détails sur les estimateurs utilisés et la modélisation réalisée.

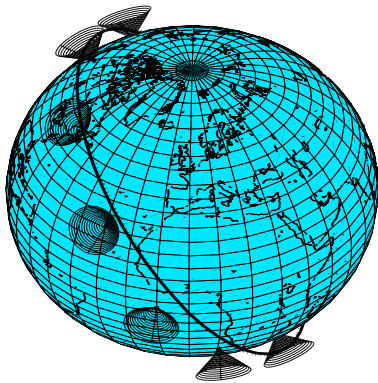


FIG. 16.12 – *Cas sans erreurs : Vue 3D*

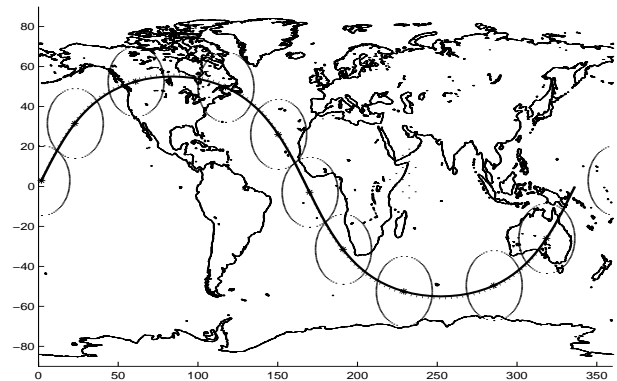


FIG. 16.13 – *Cas sans erreurs : Vue 2D*

Par ailleurs, on constate que parmi les constellations définies à ce jour, les bonnes propriétés sont parfois liées à la collaboration étroite d'un sous ensemble de satellites ([64]). Par exemple, dans son article [37], E. Frayssinhes étudie des constellations dédiées à la navigation et présente un groupement de trois satellites évoluant ensemble et gardant une bonne géométrie relative. De même, une étude réalisée au cours de ce travail doctoral ([53]) a permis de mettre en évidence l'utilité

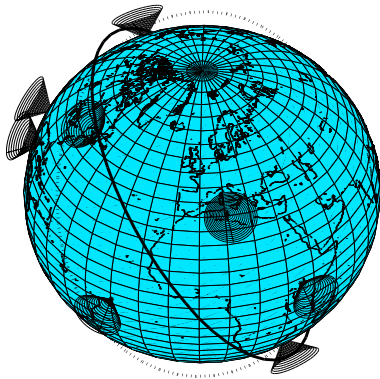


FIG. 16.14 – *Points aberrants : Vue 3D*

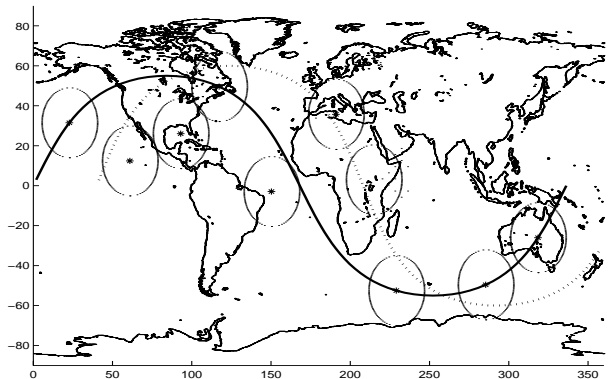


FIG. 16.15 – *Points aberrants : Vue 2D*

de doublets de satellites au sein d'une constellation. En allant un peu plus loin, certaines études ([10],[18]) en navigation ont montrées que l'utilisation de constellations hybrides permettait d'accroître les performances : chaque sous constellation compense les manques de l'autre (constellation *MEO* complétée par des satellites *GEO*). Enfin, un certain nombre de constellation (Skybridge, Globalstar [78]) sont composées de deux sous-constellations de Walker et ont de meilleures propriétés qu'une constellation de Walker comportant le même nombre de satellite.

Lors du déroulement de l'algorithme (algorithme 13.2 page 180), le niveau métaheuristique (niveau haut de l'algorithme), gère l'ajout et le retrait d'un satellite. De manière générale : lorsqu'un satellite est identifié comme ne fournissant pas de service suffisant, il est supprimé ; Lorsque qu'un satellite est identifié comme fournissant de bons services, ses propriétés servent pour initialiser un nouveau satellite (à travers la classe d'orbite). L'inconvénient d'un tel procédé est que l'on attribut les bonnes performances de la constellation à des satellites indépendamment les uns des autres.

Nous proposons donc comme seconde perspective d'étendre la modélisation présentée dans la partie IV afin de détecter et gérer les sous-constellations. On peut définir, au même titre que l'*ODB* regroupe des classes d'orbites, une base de données composée de configurations (de petite taille). L'algorithme de haut niveau pourrait ainsi gérer l'ajout et le retrait de plusieurs satellites en même temps. Le concept semble intéressant mais une étude est nécessaire avant de pouvoir conclure. L'ajout de configurations rend plus complexe le voisinage d'une solution. Le schéma des configurations que l'on stocke dans la base reste à définir ainsi que les paramètres les définissant (écart angulaire, différence d'altitude, ...).

Partie VI

Annexe

Annexe A

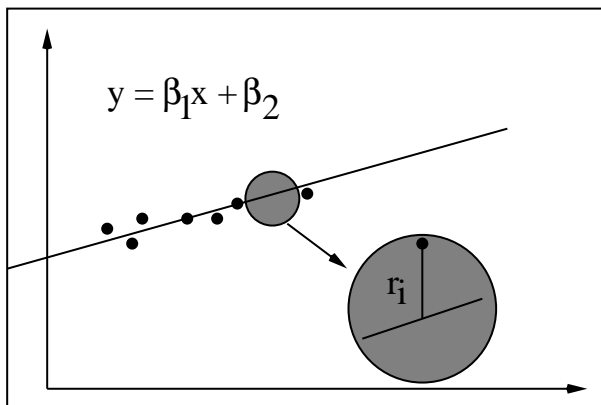
Estimation de paramètres

A.1 Introduction

Nous allons présenter brièvement les techniques d'estimation au sens des moindres carrés ([92], [66]) en nous limitant au cas non-linéaire (algorithme de type Newton [34]). Ce choix est lié au cas d'application qui nous intéresse, les équations liants les paramètres à estimer et les mesures extraites du problème étant non-linéaires ([66]).

A.2 Problème de régression aux moindres carrés ordinaires (OLS)

On cherche à estimer la valeur du paramètre $\beta \in \mathbb{R}^p$. Ce vecteur β intervient dans une relation du type $y = f(x, \beta)$ ($x \in \mathbb{R}^n, y \in \mathbb{R}^q$), décrivant une transformation permettant de modéliser l'obtention de données de sortie $y \in \mathbb{R}^q$ en fonction de stimuli d'entrée $x \in \mathbb{R}^n$ et de β inconnu.



Pour pouvoir évaluer β , on réalise un nombre m d'expériences (de mesures) liants un ensemble d'entrées $\tilde{x}_i (i \in [1, m])$ à un ensemble de sorties $\tilde{y}_i (i \in [1, m])$.

On cherche ensuite la valeur de β solution de :

$$(OLS) \begin{cases} \text{Min} \frac{1}{m} \sum_{i=1}^m r_i(\beta)^2 \\ \beta \end{cases} \quad (A.1)$$

avec $r_i(\beta) = \|\tilde{y}_i - f(\tilde{x}_i, \beta)\|$

La figure A.1 illustre le cas particulier où $p = 2$, $n = 1$, $q = 1$ et $f(x, \beta) = \beta_1 \cdot x + \beta_2$.

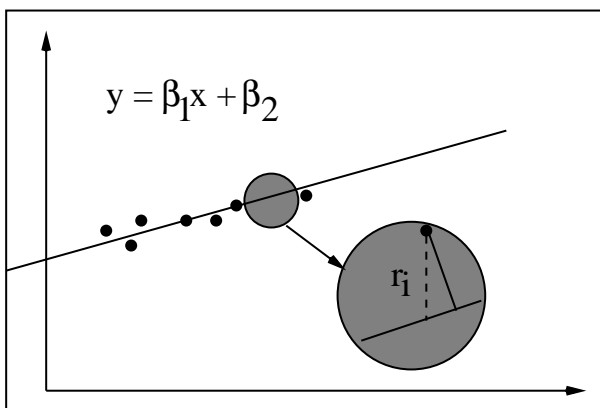
FIG. A.1 – Estimation moindres carrés ordinaires

En pratique, si f est non linéaire par rapport à

β , on utilise une méthode itérative de type Newton pour améliorer une estimation initiale β_0 .

$$x = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} \rightarrow \beta = \begin{bmatrix} \beta_1 \\ \dots \\ \beta_p \end{bmatrix} \rightarrow y = f(x, \beta) = \begin{bmatrix} f_1(x, \beta) \\ \dots \\ f_q(x, \beta) \end{bmatrix} \quad (\text{A.2})$$

A.3 Régression non linéaire en distance orthogonale (ODR)



Ce deuxième estimateur diffère du premier par l'introduction de variables d'erreurs (δ) sur les données d'entrée. Le nouveau problème posé est donc :

$$(\text{ODR}) \begin{cases} \text{Min} \frac{1}{n} \sum_{i=1}^n r_i(\beta, \delta_i)^2 \\ \beta, \delta \end{cases} \quad (\text{A.3})$$

avec $r_i^2 = \epsilon_i \omega \epsilon_i^t + \delta_i \omega_\delta \delta_i^t$ et $\epsilon_i = \tilde{y}_i - f(\tilde{x}_i + \delta_i, \beta)$. La figure A.2 reprend le cas particulier précédent pour un estimateur ODR.

FIG. A.2 – Estimation en distance orthogonale

A.4 Points aberrants et points levier

Un inconvénient supplémentaire peut apparaître lorsque l'on considère que certaines mesures peuvent être entachées d'erreurs (C'est souvent le cas dans la réalité). Ces erreurs peuvent conduire à l'apparition de points aberrants parmi les mesures réalisées. Un point aberrant (figure A.3) est un point dont la valeur s'écarte du reste du nuage de point constitué par les mesures. L'influence d'un point aberrant sur la qualité de la solution est accrue lorsque celui-ci est en plus un point levier (figure A.3).

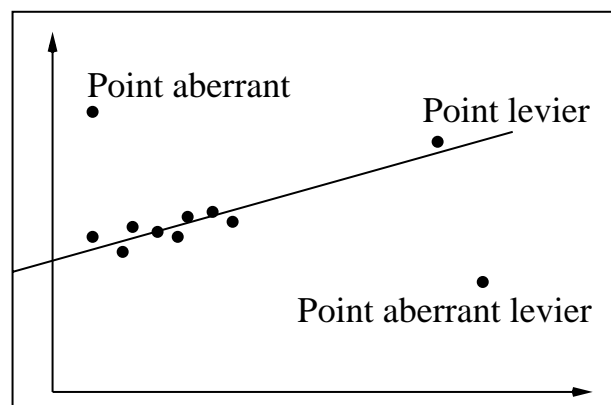


FIG. A.3 – Points aberrants et points levier

A.5 Estimation robuste : les M-Estimeurs

Principe

Pour résoudre les problèmes liés aux points aberrants, il existe un ensemble d'estimateurs robustes ([33], [72]) qui permettent de réduire l'influence de ces points.

Les M-estimateurs

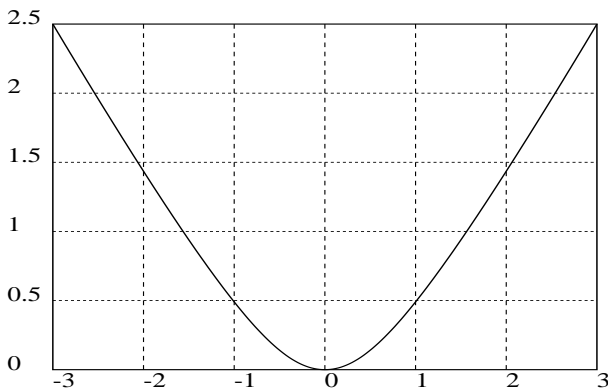


FIG. A.4 – Fonction de Huber avec $a=1$

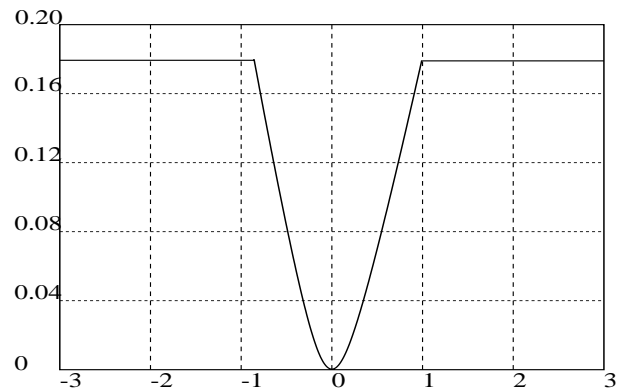


FIG. A.5 – Fonction de Tuckey avec $a=1$

Pour construire un estimateur aux moindres carrés résistant aux points aberrants, on peut par exemple considérer les M-estimateurs. Le problème devient :

$$(M - estimateurs) \left\{ \begin{array}{l} \text{Min} \frac{1}{n} \sum_{i=1}^n \rho(r_i) \\ \beta \end{array} \right. \quad (\text{A.4})$$

la fonction ρ est appliquée aux résidus r_i de manière à éliminer l'influence des valeurs aberrantes. Deux fonctions couramment utilisées sont présentées dans les figures A.4, A.5. ρ doit être adaptée à la loi de probabilité des erreurs afin de respecter le principe du maximum de vraisemblance [66].

A.6 Application à l'estimation d'orbites

Principe

Nous proposons ici d'estimer les 6 paramètres orbitaux décrivant le comportement d'un satellite en fonction des positions au sol que l'on souhaite qu'il couvre. Nous rappelons que les paramètres orbitaux dérivent au cours du temps (chapitre 3) mais que la connaissance de leur valeur ($\beta_r = (a_r, e_r, i_r, \omega_r, \Omega_r, M_r)$) à une date de référence t_r , suffit à connaître leur valeur à tout instant t .

Nous allons donc chercher à estimer ce jeu de paramètre β_r .

Premier modèle : modèle géométrique

Le premier modèle que nous présentons est uniquement basé sur la position des points que l'on souhaite couvrir. On ne prend pas en compte le paramètre temporel. On est donc dans le cas suivant : $n = 2$, $p = 6$, $q = 1$ et $y = f(x, \beta_r) = 0$. Les données en entrée sont les coordonnées longitudinales des points d'intérêt.

$$x = \begin{bmatrix} Lon \\ Lat \end{bmatrix} \rightarrow \beta_r = \begin{bmatrix} a_r \\ e_r \\ i_r \\ \omega_r \\ \Omega_r \\ M_r \end{bmatrix} \rightarrow y = f(x, \beta_r) = Min_t (\| \begin{bmatrix} Lon - f_1(t, \beta_r) \\ Lat - f_2(t, \beta_r) \end{bmatrix} \|) = 0 \quad (A.5)$$

$f_1(t, \beta_r)$ et $f_2(t, \beta_r)$ expriment les coordonnées longitudinales à la date t du point sous satellite, dont les paramètres orbitaux à la date t_r sont β_r . La valeur de y est donc la distance minimale entre la trace du satellite et le point d'intérêt x ($y = 0$ dans le cas idéal).

Pour notre problème d'estimation, on cherche la trace au sol qui minimise la distance (au sens *OLS* ou *ODR*) entre celle-ci et l'ensemble des m points d'intérêt présentés en entrée de l'estimateur.

Il faut par ailleurs noter que l'estimateur nécessite un jeu de paramètres initiaux (β_0) pour être mis en oeuvre.

Deuxième modèle : modèle cinématique

On intègre maintenant la notion de temps dans les entrées du système. En d'autres termes, le satellite doit être en visibilité d'un point d'intérêt P_i à un instant t_i .

On obtient le cas particulier suivant : $n = 3$, $p = 6$, $q = 1$ et $y = f(x, \beta_r) = 0$.

$$x = \begin{bmatrix} Lon \\ Lat \\ t \end{bmatrix} \rightarrow \beta_r = \begin{bmatrix} a_r \\ e_r \\ i_r \\ \omega_r \\ \Omega_r \\ M_r \end{bmatrix} \rightarrow y = f(x, \beta_r) = \| \begin{bmatrix} Lon - f_1(t, \beta_r) \\ Lat - f_2(t, \beta_r) \end{bmatrix} \| = 0 \quad (A.6)$$

Différence entre le modèle géométrique et le modèle cinématique

Dans le modèle géométrique on cherche à minimiser la distance du point d'intérêt $P = [Lon, Lat]$ à la trace du satellite (figure A.6) alors que dans le modèle cinématique on cherche à minimiser la distance du point d'intérêt $P(t) = [Lon, Lat]$ à la position du point sous satellite à l'instant t associé à ce même point (figure A.7).

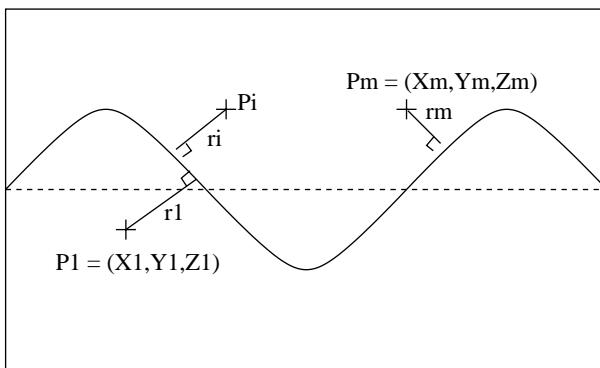


FIG. A.6 – *Modèle géométrique*

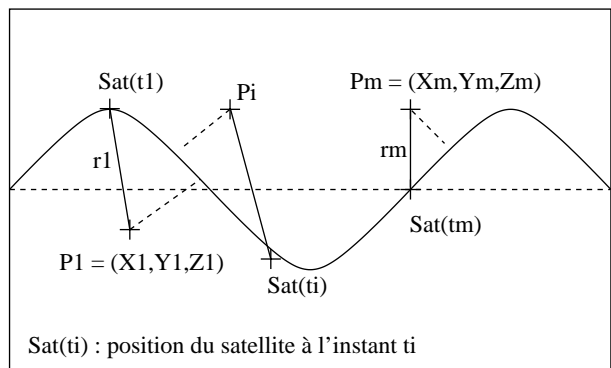


FIG. A.7 – *Modèle cinématique*

Application de l'estimation en cours d'algorithme

Cette estimation ne peut pas être appliquée en cours d'algorithme car à ce stade on dispose de trop de points d'intérêt non satisfaits et étant donné qu'une orbite ne peut satisfaire qu'un sous ensemble restreint de ces points on se trouve donc en présence d'un grand nombre de points aberrants en entrée du processus d'estimation.

Annexe B

les outils

B.1 Présentation

L'algorithme que nous avons décrit au cours de la partie IV a donné lieu au développement de plusieurs logiciels complexes permettant à la fois de paramétrer l'algorithme (problème traité, *ODB*, algorithme de niveau numérique, algorithme de niveau métaheuristique, ...), de visualiser les résultats de celui-ci (paramètres orbitaux de la solution, configuration trouvée,...) ainsi que le parcours réalisé depuis l'initialisation (courbes de valeurs, configurations visitées, historique de la recherche itération par itération, ...). Par ailleurs, l'interface développée permet une interaction poussée entre l'utilisateur et l'algorithme (points d'arrêt, visualisation intermédiaire, ...).

Les deux environnements de développement utilisés sont : *Visual C++* et *Matlab*. Le premier servant essentiellement à la partie algorithmique et le second à la visualisation.

Nous allons présenter au cours de cette annexe la structure logiciel générale ainsi que des illustrations, non exhaustives, des interfaces et fonctionnalités des logiciels.

B.2 Illustration

Voici quelques exemples de ce qu'il est possible de faire avec l'outils :

- visualiser (figure B.4) et définir l'*ODB* (1 figure B.2)
- visualiser la solution courante (2 figure B.2)
- visualiser (figure B.3) et définir les zones d'intérêt (3 figure B.2)
- définir le critère utilisé (4 figure B.2)
- paramétrer la simulation (5 figure B.2)
- paramétrer l'algorithme (6 figure B.2), la recherche tabou (7 figure B.2), la recherche de plus grande pente (8 figure B.2)

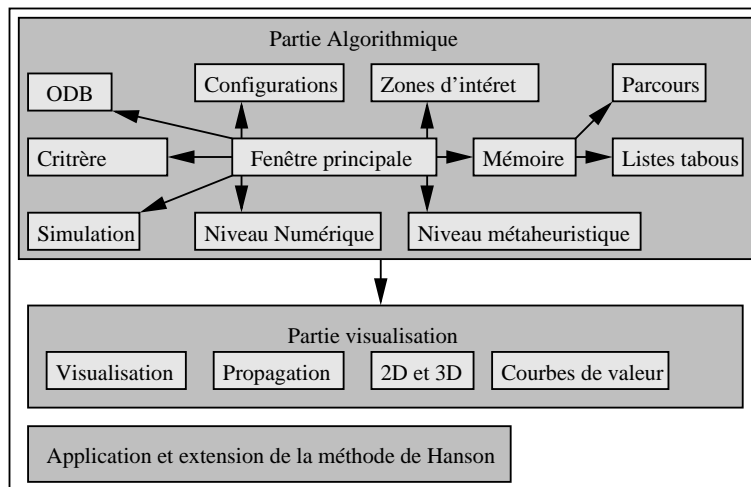


FIG. B.1 – Structure générale

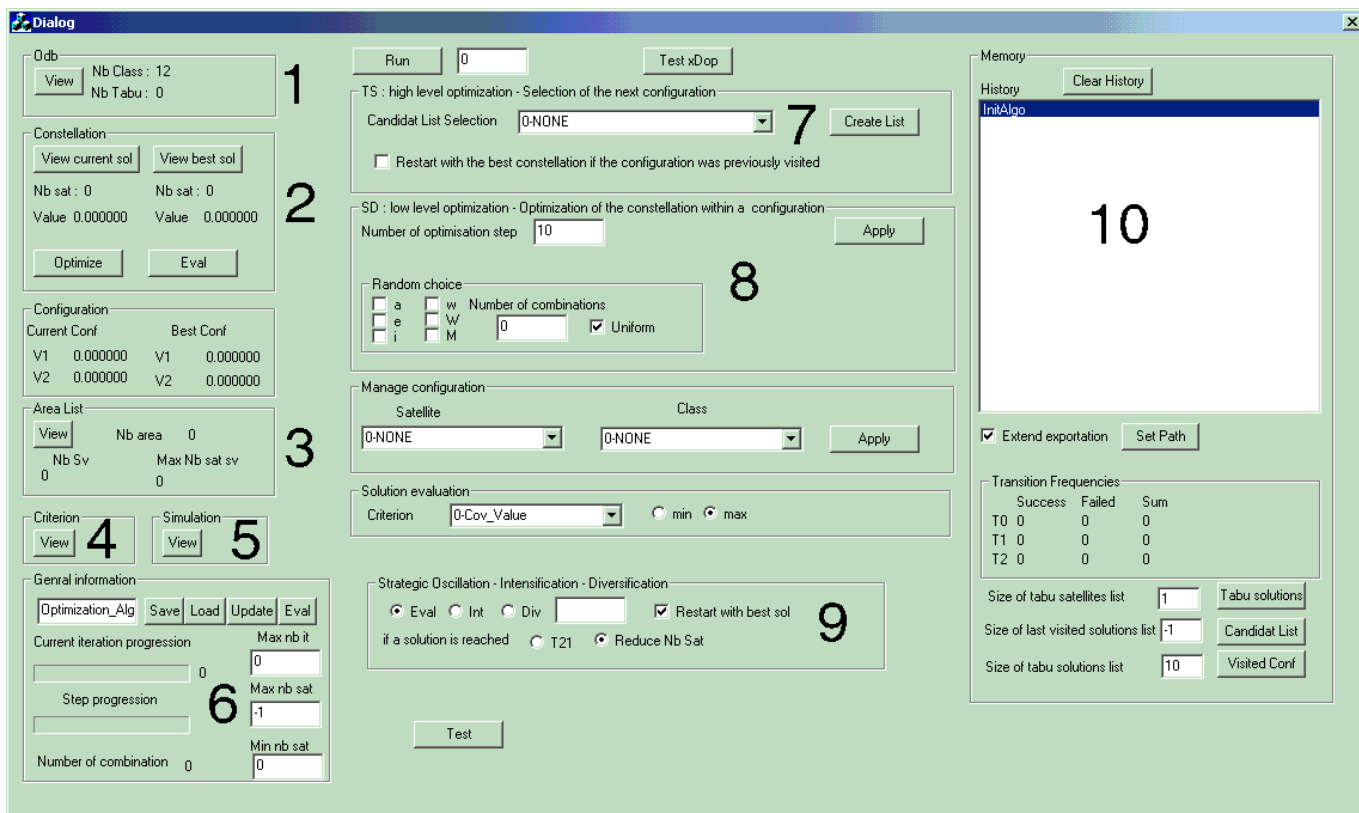


FIG. B.2 – Fenêtre principale : gestion de l'algorithme

- paramétrer l'algorithme les oscillations stratégiques (9 figure B.2)
- visualiser l'état de la mémoire (10 figure B.2)
- visualiser et modifier (figure B.6) la trace au sol associée à une classe (1 figure B.4).
- visualiser la classe associée à un satellite (1 figure B.5).

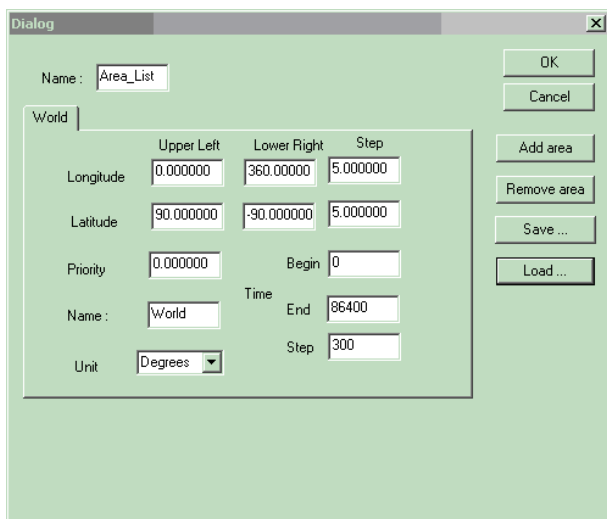


FIG. B.3 – Zones d'intérêt



FIG. B.4 – Classes de l'Odb

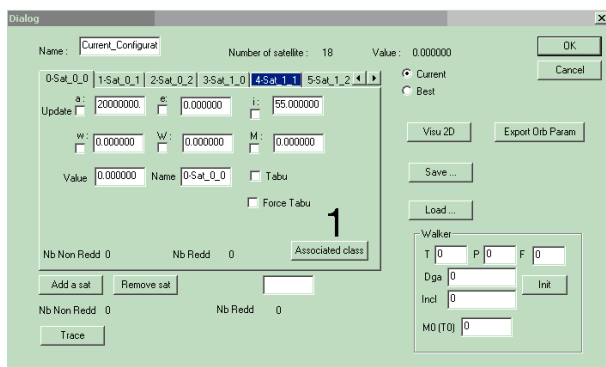


FIG. B.5 – La configuration courante

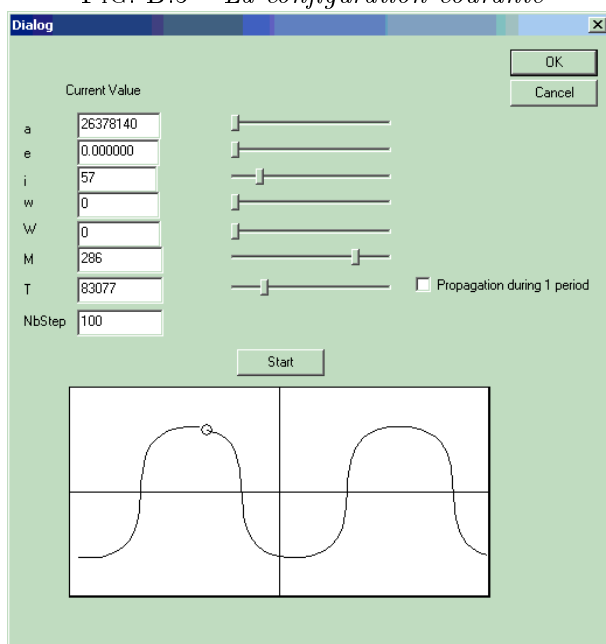


FIG. B.6 – Trace associée à une classe

B.3 Méthode de Hanson

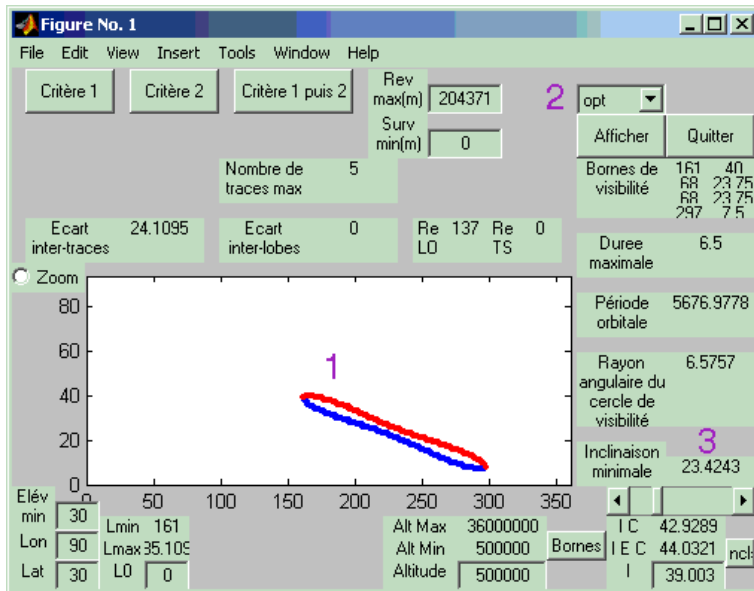


FIG. B.7 – Méthode de Hanson

La méthode de Hanson a donné lieu à un outils indépendant développé sous Matlab (figure B.7). Il permet entre autre de visualiser les lobes (1), de sélectionner le type d'instrument (radar ou optique, 2), de déterminer l'inclinaison critique (3).

Bibliographie

- [1] S. Abbondanza and al. Design of meo constellations for galileo : Towards a "design to cost" approach. In *50th International Astronautical Federation Congress*, 1999.
- [2] S. Abbondanza and al. Optinav : An efficient multi-level approach for the design of navigation constellations - application to galileo. In *GNSS*, 1999.
- [3] Adams. Circular polar constellations providing continuous single or multiple coverage above a specified latitude. In *Journal Of The Astronautical Sciences*, 1987.
- [4] Jean-Marc Alliot. Techniques d'optimisation stochastique appliquées aux problèmes de trafic aérien. In *Mémoire d'HDR, Enac*, 1996.
- [5] A.H. Ballard. Rosette constellations of earth satellite. In *IEEE Transactions On Aerospace And Electronic Systems*, 1980.
- [6] Bandler and al. Fuzzy power sets and fuzzy implication soperators. In *Fuzzy Sets and Systems*, 1980.
- [7] H. Baranger and al. Global optimization of gps type satellite constellations. In *42th International Astronautical Federation Congress*, 1991.
- [8] Thomas Bäck and al. An overview of evolutionary algorithms for parameter optimization. In *Evolutionary Computation*, 1993.
- [9] T. W. Beech and al. A study of three satellite constellation design algorithms. 1999.
- [10] A. Benhallam and al. Contribution of hybridization to the performance of satellite navigation systems. 1999.
- [11] D.C. Beste. Design of satellite constellations for optimal continuous coverage. In *IEEE Transactions On Aerospace And Electronic Systems*, 1978.
- [12] J. Blamont. Towards an integrated space system. In *48th International Astronautical Federation Congress*, 1997.
- [13] C. Bourga and al. Performance of a low earth orbit navigation system : Error model, impact of doppler measurements and comparison with gps. In *GNSS*, 1997.
- [14] C. Brochet and al. Planification et maintien a poste de constellations de satellites : Modélisation, optimisation, simulation. In *LAAS-CNRS*, 1998.

- [15] C. Brochet and al. A multiobjective optimization approach for the design of walker constellation. In *50th International Astronautical Federation Congress*, 1999.
- [16] R. G. Brown. A baseline gps raim scheme and a note on the equivalence of three raim methods. In *Journal Of The Institute Of Navigation*, 1992.
- [17] R. Cerf. Une théorie asymptotique des algorithmes génétiques. In *PhD Univ. de Montpellier II France*, 1994.
- [18] J. Chenebault. Ines an inovative european navigation system relying on a low earth orbit constellation. 1998.
- [19] P. Christopher. Constellations for high elevation angles. In *48th International Astronautical Federation Congress*, 1997.
- [20] CNES. Techniques et technologies des véhicules spatiaux. In *Cépadués Editions*, 1989.
- [21] H. Curry. The method of steepest descent for nonlinear minimization problems. In *Quart. Appl. Math.*, 1944.
- [22] D. Diekelman. Design guidelines for post 2000 constellations. In *48th International Astronautical Federation Congress*, 1997.
- [23] Drain. Satellite constellations for optimal northern hemisphere coverage. In *9th DARPA Strategic Space Symposium*, 1983.
- [24] Drain. A common period four-satellite continuous global coverage constellation. In *Journal Of Guidance, Control, And Dynamics*, 1987.
- [25] Drain. A six-satellite continuous global double coverage constellation. In *Astrodynamics Conference*, 1987.
- [26] Drain. Continuous global n-tuple coverage with $2n+2$ satellites. In *Astrodynamics Conference*, 1989.
- [27] J. Drain. Optimization of the ellipso and ellipso-2g personal communication systems. 1996.
- [28] Dubois and al. A class of fuzzy measures based on triangle inequalities. In *Int. J. Gen. Sys.*, 1983.
- [29] F. Dufour and al. Constellation design optimization with a dop based criterion. In *14th Int. Symposium On Space Fligt Dynamics*, 1995.
- [30] F. Dufour and al. A multistage approach to design and optimize a communication satellite constellaiton. In *50th International Astronautical Federation Congress*, 1999.
- [31] N. Durand. Optimisation de trajectoires pour la résolution de conflits aériens en route. 1996.
- [32] T.A. Ely. Satellite constellation design for zonal coverage using genetic algorithms. In *Space Flight Mechanics Meeting*, 1998.
- [33] PT. Boggs et and al. Orthogonal distance regression. In *Contemporary Mathematics*, 1990.
- [34] R. Fletcher. Practical methods of optimization. In *Wiley*, 1987.

- [35] D. B. Fogel. An introduction to simulated evolutionary optimization. In *IEEE*, 1994.
- [36] E. Frayssinhes. Investigating new satellite constellation geometries with genetic algorithms. In *American Institute Of Aeronautics and Astronautics*, 1996.
- [37] E. Frayssinhes and al. Designing clusters of satellites for radiolocalisation purposes. In *Astrodynamics Conference*, 1995.
- [38] E. Frayssinhes and al. Mission analysis of clusters of satellites. In *46th International Astronautical Federation Congress*, 1995.
- [39] F. Glover. Candidate list strategies and tabu search. 1989.
- [40] F. Glover. Tabu search part i. In *ORSA Journal On Computing*, 1989.
- [41] F. Glover. Tabu search part ii. In *ORSA Journal On Computing*, 1990.
- [42] F. Glover. Scatter search and star-paths : Beyond the genetic metaphor. 1993.
- [43] F. Glover. Genetic algorithms and scatter search : unsuspected potentials. In *Statistics And Computing*, 1994.
- [44] F. Glover. Genetic algorithms and scatter search : Unsuspected potentials. In *Statistics And Computing*, 1994.
- [45] F. Glover and al. Finite convergence of tabu search. In *MIC conference*, 2001.
- [46] F. Glover and M. Laguna. Tabu search. In *Kluwer Academic Publishers*, 1997.
- [47] GNSS. Performances of leo navigation systems : Error model, impact of doppler measurements and comparison with gps. In *GNSS*, 1988.
- [48] D.G. Goldberg. Genetic algorithms in search, optimization and machine learning. In *Addison Wesley*, 1989.
- [49] E. Grandchamp. Bibliographie sur les méthodes d'optimisation. In *Rapport Interne Alcatel Space Industries*, 1998.
- [50] E. Grandchamp. Optimisation de constellations de satellites : Première approche. In *Rapport Interne Alcatel Space Industries*, 1998.
- [51] E. Grandchamp. Etude de flextool partie i, ii, iii et iv. In *Rapport Interne Alcatel Space Industries*, 1999.
- [52] E. Grandchamp. Etude de la méthode de hanson. In *Rapport Interne Alcatel Space Industries*, 1999.
- [53] E. Grandchamp. Etude des doublets de satellites. In *Rapport Interne Alcatel Space Industries*, 1999.
- [54] E. Grandchamp. Modélisation mathématique des critères. In *Rapport Interne Alcatel Space Industries*, 1999.
- [55] E. Grandchamp. Classification des orbites. In *Rapport Interne Alcatel Space Industries*, 2000.

- [56] E. Grandchamp. Modélisation et algorithme de résolution. In *Rapport Interne Alcatel Space Industries*, 2000.
- [57] E. Grandchamp. Optimisation de constellations par approche géométrique. In *Rapport Interne Alcatel Space Industries*, 2000.
- [58] E. Grandchamp and V. Charvillat. Integrating orbit database and metaheuristics to design satellite constellation. In *Proceedings of ICAI'2000*, pages 733–739, 2000.
- [59] E. Grandchamp and V. Charvillat. Satellite constellations optimization with metaheuristics. In *Proceedings of Euro'2000*, page 139, 2000.
- [60] E. Grandchamp and V. Charvillat. Metaheuristics to design satellite constellations. In *Proceedings of MIC'2001*, 2001.
- [61] J. Grasmeyer. Application of genetic algorithm with adaptive penalty functions to airfoil design. In *48th International Astronautical Federation Congress*, 97.
- [62] P. Hansen and al. Variable neighborhood search : Principles and applications. In *Les cahiers du GERAD*, 1998.
- [63] J.M. Hanson and al. Designing good partial coverage satellite constellations. In *The Journal Of The Astronautical Sciences Vol 40*, 1992.
- [64] Hastings and al. The generalized analysis of distributed satellite systems. 1995.
- [65] J.H. Holland. Adaptation in natural and artificial systems. In *University Of Michigan Press*, 1975.
- [66] P.J. Huber. Robust statistics. In *John Wiley and Sons*, 1981.
- [67] P. Janniere and al. Satellite constellations for future global navigation systems : Design and comparison of solutions. In *45th International Astronautical Federation Congress*, 1994.
- [68] P. Janniere and al. Dimensionnement de constellations pour les futurs systèmes civils de navigation par satellites. In *SEE*, 1995.
- [69] P. Jannière and al. The use of genetic algorithms for the design of future satellite navigation systems. In *ASTAIR*, 1994.
- [70] D. S. Johnson and al. Optimization by simulated annealing : An experimental evaluation; part i. In *Operational Research*, 1989.
- [71] D. S. Johnson and al. Optimization by simulated annealing : An experimental evaluation; part ii. In *Operational Research*, 1989.
- [72] Åke Björck. Numerical methods for least squares problems. In *SIAM*, 1996.
- [73] C. W. Kelley. Specifications to obtain category 1 precision approach worldwide. In *ION*, 1999.
- [74] K. Kimura and al. Optimum constellations using elliptical inclined synchronous orbits for mobile and fixed satellite communications. 1999.

- [75] T. Lang and J.M. Hanson. Orbital constellations which minimize revisit time. In *Astrodynamics Conference*, 1983.
- [76] T. J. Lang. Symmetric circular orbit satellite constellations for continuous global coverage. In *Astrodynamics Conference*, 1987.
- [77] T. J. Lang. A comparison of satellite constellations for continuous global coverage. In *48th International Astronautical Federation Congress*, 1997.
- [78] E. Lansard and al. Global design of satellite constellations: A multi-criteria performance comparison of classical walker patterns and new design patterns. In *47th International Astronautical Federation Congress*, 1996.
- [79] E. Lansard and al. Operational availability. In *48th International Astronautical Congress*, 1997.
- [80] E. Lansard and al. Operational availability of satellite constellations. In *48th International Astronautical Federation Congress*, 1997.
- [81] E. Lasserre. Sur l'optimisation de la mise et du maintien a poste de constellations de satellites. In *LAAS-CNRS*, 1998.
- [82] R. D. Lüders. Satellite networks for continuous zonal coverage. In *ARS Journal*, 1961.
- [83] V. Martinot and al. Deployment and maintenance of satellite constellations. In *49th International Astronautical Federation Congress*, 1998.
- [84] M. Bello Mora and al. Orion - a constellation mission analysis tool. In *48th International Astronautical Federation Congress*, 1997.
- [85] Bradford W. Parkinson. Gps : theory and applications volume i. 1993.
- [86] Greistorfer Peter. On the algorithmic design in heuristic search. In *Euro conference*, 2000.
- [87] G. Rauwolf and al. Near optimal solar sail trajectories generated by a genetic algorithm. In *Astrodynamics Conference*, 1999.
- [88] H. Renault. Constellation studies for future navigation system. In *48th International Astronautical Federation Congress*, 1997.
- [89] J.T. Richardson and al. Some guidelines for genetic algorithm with penalty functions. 1993.
- [90] Rider and al. Analytic design of satellite constellations for zonal earth coverage using inclined circular orbits. In *Journal Of The Astronautical Sciences*, 1986.
- [91] M. Romay-Merino and al. Design of high performance and cost efficient constellations for gnss-2. In *GNSS*, 1998.
- [92] P.J Rousseeuw and al. Robust regression and outlier detection. In *John Wiley and Sons*, 1987.
- [93] A.E. Smith and al. Genetic optimization using a penalty function. 1992.

- [94] E. Smith and al. An operational approach for generating near-optimal station keeping strategies via parallel genetic algorithms. In *Astrodynamics Conference*, 1999.
- [95] M. Srinivas and al. Genetic algorithms : A survey. In *IEEE*, 1994.
- [96] N. Sultan and al. Effect of mobile satellite system design on cost / revenues of total earth and space segments. In *47th International Astronautical Federation Congress*, 1996.
- [97] Fuhan Tian. The influence of crossover operator on the diversity in genetic algorithms. In *IC-AI conference*, 2000.
- [98] M. Vasquez and J-K. Hao. A heuristic approach for antenna positioning in cellular networks. In *Journal Of Heuristics*, 2000.
- [99] M. Vasquez and J-K. Hao. A "logic-constraint" knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. In *Journal Of Computational Optimization And Applications*, 2000.
- [100] J.G Walker. Circular orbit patterns providing whole earth coverage. In *Royale Aircraft Establishment Technical Report*, 1970.
- [101] JS. Wu and al. Non precision approach availability sensitivity to gps satellite constellation configuration. In *ION*, 1999.
- [102] Zadeh and al. Fuzzy sets. In *Information and Control*, 1965.
- [103] M. Zbigniew. Genetic algorithms + data structures = evolution programs. In *Springer-Verlag*, 1994.
- [104] F. Zhang and al. Optimization design method and its application to satellite constellation system design. 2001.
- [105] Z. Zhang. Parameter estimation techniques : A tutorial with application to conic fitting. In *Image And Vision Computing*, 1997.
- [106] F. Zwolska and al. Optimization of a navigation constellation design based on operational availability constraint. In *GNSS*, 1998.